

# DUNE 数据查询与可视化

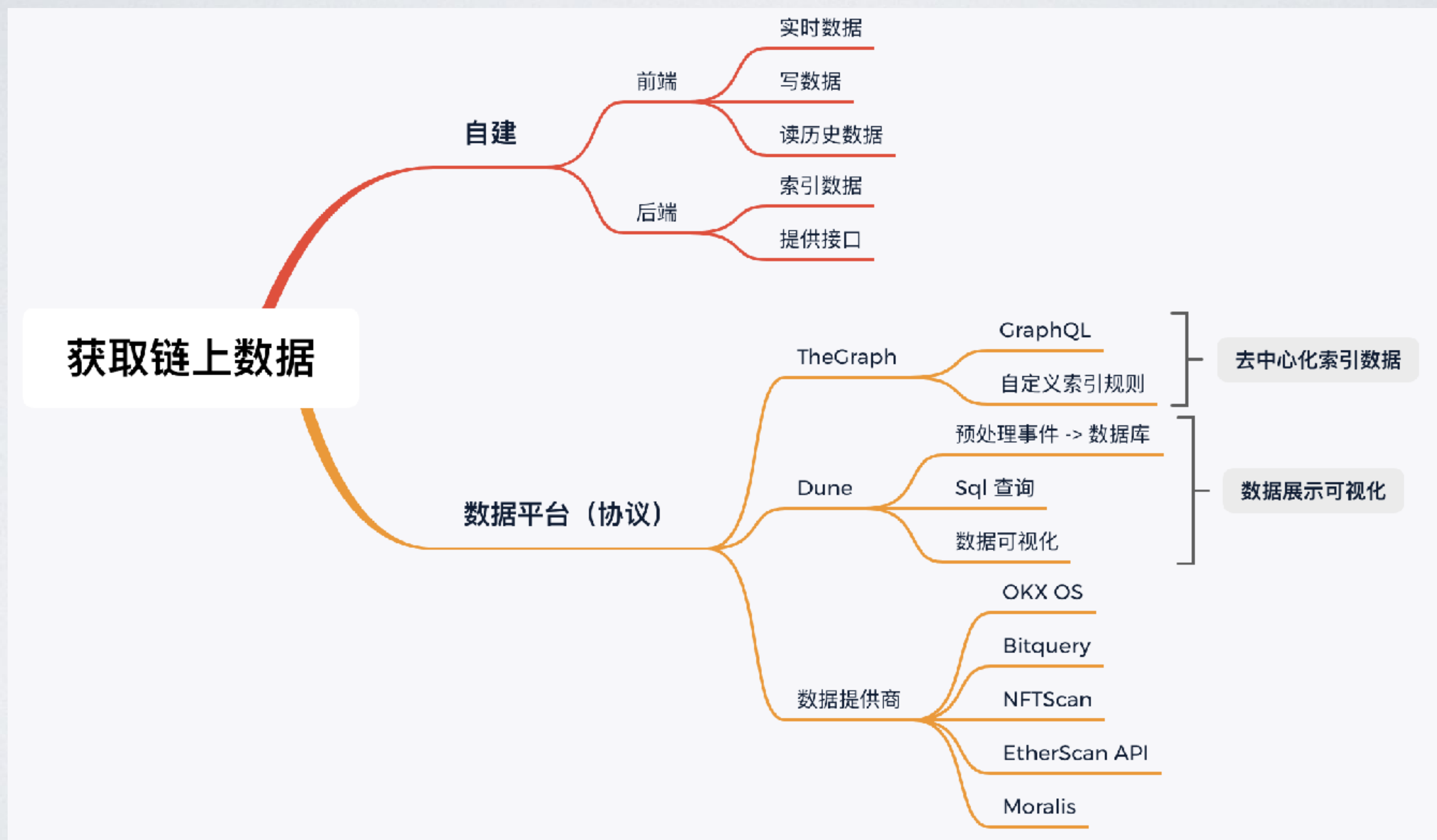
Tiny 熊

# 要点

- 理解 Dune 功能以及如何工作的
- 基础查询：ETH 交易量、以太坊价格、ETH 交易价值等...
- 为 PEPE 创建仪表盘(持币用户数数据以及交易活跃数据)

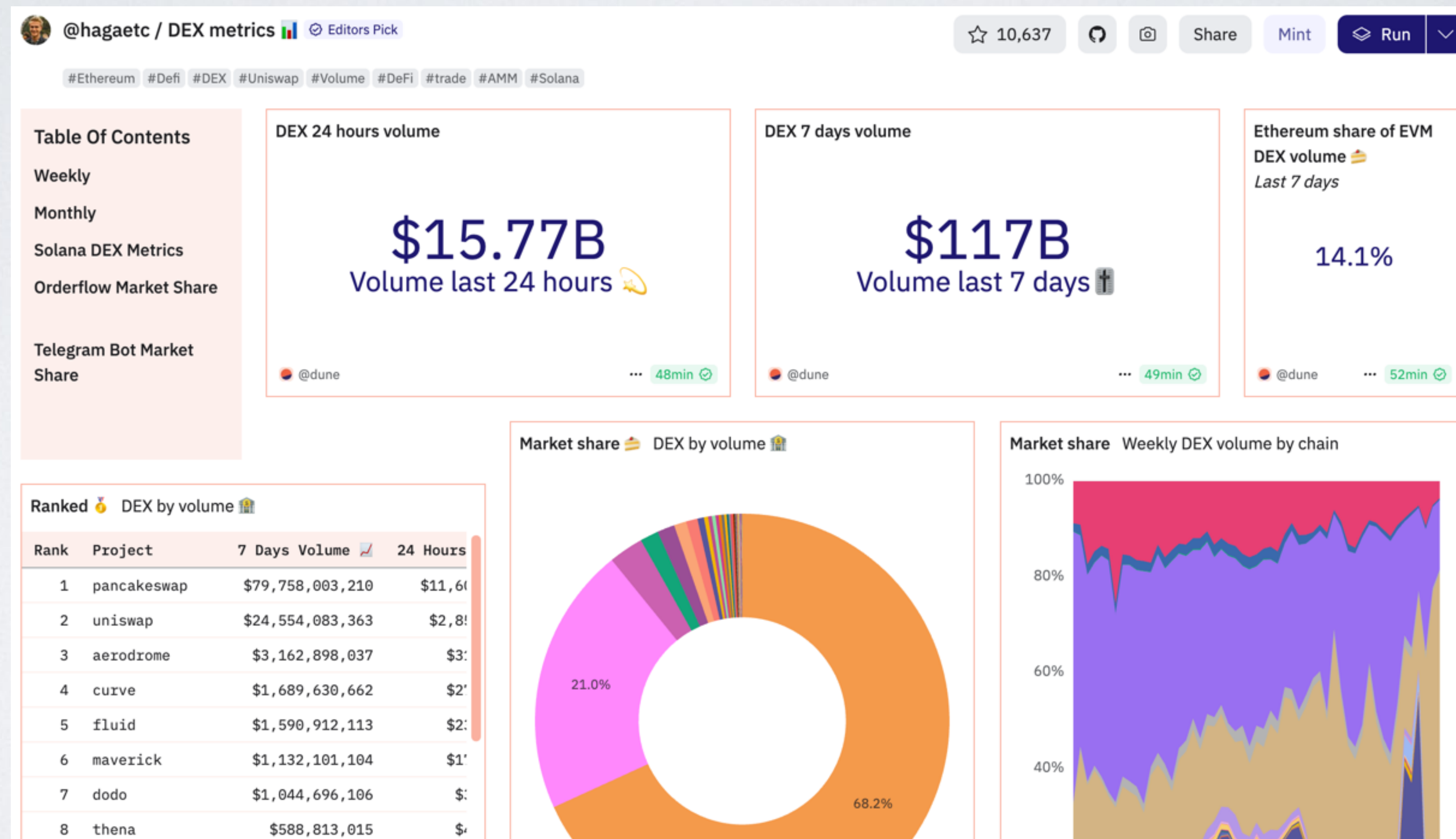


# 获取数据的方式



# DUNE

- Dune 是一款 Web 应用，任何人（具有一点 SQL 知识）都可以查询区块链数据，创建各种图表。



<https://dune.com/hagaetc/dex-metrics>



# DUNE 功能

- 提供了丰富的数据集，方便大家在数据集上构建查询。（可基于他人的查询基础上修改）
- Dune AI：将自然语言转为 SQL
- 可视化图表和仪表板将数据转化为洞见
- 可将某个具体的可视化图表分享，并嵌入到其他页面
- SIM API：通过 API 访问数据

<https://docs.dune.com/web-app/overview>

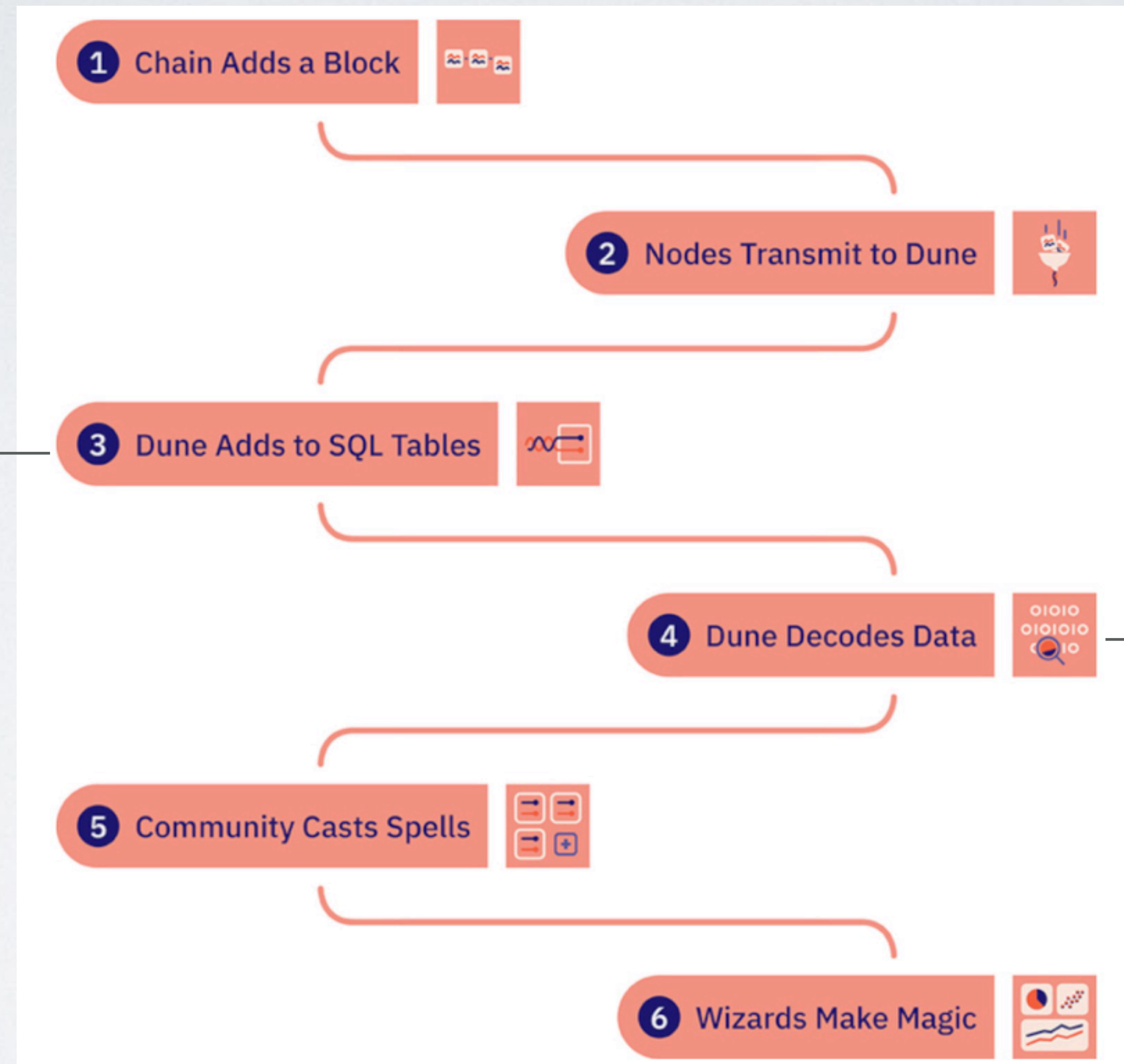


# DUNE 是如何工作的

← Raw tables Ethereum

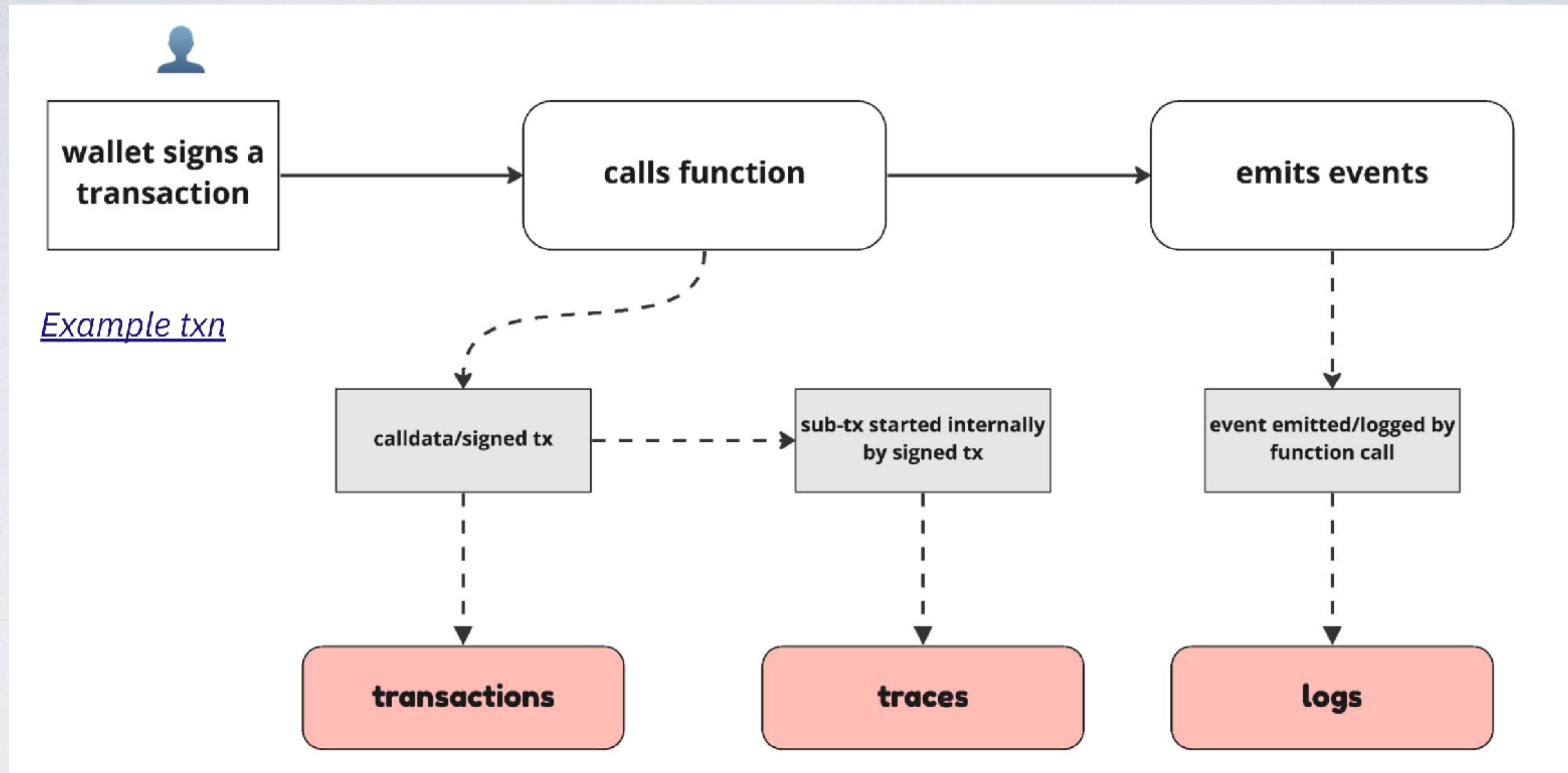
ethereum	blocks	🔍	👁	➡
ethereum	creation_traces	🔍	👁	➡
ethereum	logs	🔍	👁	➡
ethereum	logs_decoded	🔍	👁	➡
ethereum	traces	🔍	👁	➡
ethereum	traces_decoded	🔍	👁	➡
ethereum	transactions	🔍	👁	➡

各种聚合表：如 Dex、价格表



erc20

erc20	evt_Transfer	🔍	👁	➡
erc20	evt_Transfer	🔍	👁	➡
erc20	evt_Transfer	🔍	👁	➡
erc20	evt_Transfer	🔍	👁	➡
ovm1	erc20_balances	🔍	👁	➡
tokens	erc20	🔍	👁	➡
tokens	erc20	🔍	👁	➡
tokens	erc20_bridged_mapping	🔍	👁	➡



交易

ethereum.transactions

ethereum.traces

ethereum.logs



# DUNE 数据分类

Data explorer

Search by dataset name, contract address...

Curated data

- Asset tracking
- Trading
- Labels and utility tables
- All curated data

My data

- Materialized views
- Uploads
- Favorites
- On this query

Blockchain data

- Decoded projects
- Raw blockchain data

Community data

- Curated community uploads
- Uploads

- BlockchainData
  - Raw data: EVM chains 、 Other chains
  - Decoded data: 解析后的数据
    - ethereum.contracts 记录了所有解析了的合约
- Curated(精选) Data : token balances 、 dex.trades 、 nft.trades, prices
  - 跨链数据聚合
- Community Data: lens、 flashbots ...

Ethereum

- Overview
- Raw
  - Transactions
  - Logs
  - Traces
  - Creation Traces
  - Blocks
  - Withdrawals
- Decoded
  - Overview
  - Call Tables
  - Event Logs
  - Decoded Contracts
  - logs\_decoded
  - Decoded Traces

<https://dune.com/queries>



# DUNE SQL

- DuneSQL, SQL 方言, Trino 的 Fork, 并针对链数据添加了一些函数
- SHOW FUNCTIONS; 列出了所有内置的函数:
  - EVM 0x 使用数据类型 varbinary
  - 内置解析事件/函数、JSON 操作、时间处理等
  - 可以在 DuneSQL 中做 HTTP 请求
  - ....

<https://docs.dune.com/query-engine/Functions-and-operators>

# DUNE 工作区



Data explorer

Search by dataset name, contract address...

Curated data

Asset tracking

Trading

Labels and utility tables

All curated data

My data

Materialized views

Uploads

Favorites

On this query

Blockchain data

Decoded projects

Raw blockchain data

Community data

Curated community uploads

Uploads

数据表

@xilibi2003 / Creations / New Query / Unsaved

Save

Edit SQL with prompt...

SQL 编写区

1 SELECT \* FROM ethereum.transactions ORDER BY block\_time DESC LIMIT 10

LAST RUN 1 MINUTE AGO  
LAST RUN TOOK 14 MINUTES

Run

New

Query results

Lineage

Query results New Query

结果展示区

block_time	block_number	value	gas_limit	gas_price	gas_used	max_fee_per_gas	max_priority_fee_per_gas
2025-06-01 02:43	22606951	0	97164	5000000000	48576		
2025-06-01 02:43	22606951	0	204032	4029886193	164346	5212357741	3300000000
2025-06-01 02:43	22606951	0	97164	5000000000	48588		
2025-06-01 02:43	22606951	0	46225	6000000000	41309		
2025-06-01 02:43	22606951	500000000000000000	301844	3729886193	174819	3978490220	3000000000
2025-06-01 02:43	22606951	0	8000000	3729886193	85905	33000000000	3000000000
2025-06-01 02:43	22606951	500000000000000000	275831	3729886193	117315	3978490220	3000000000

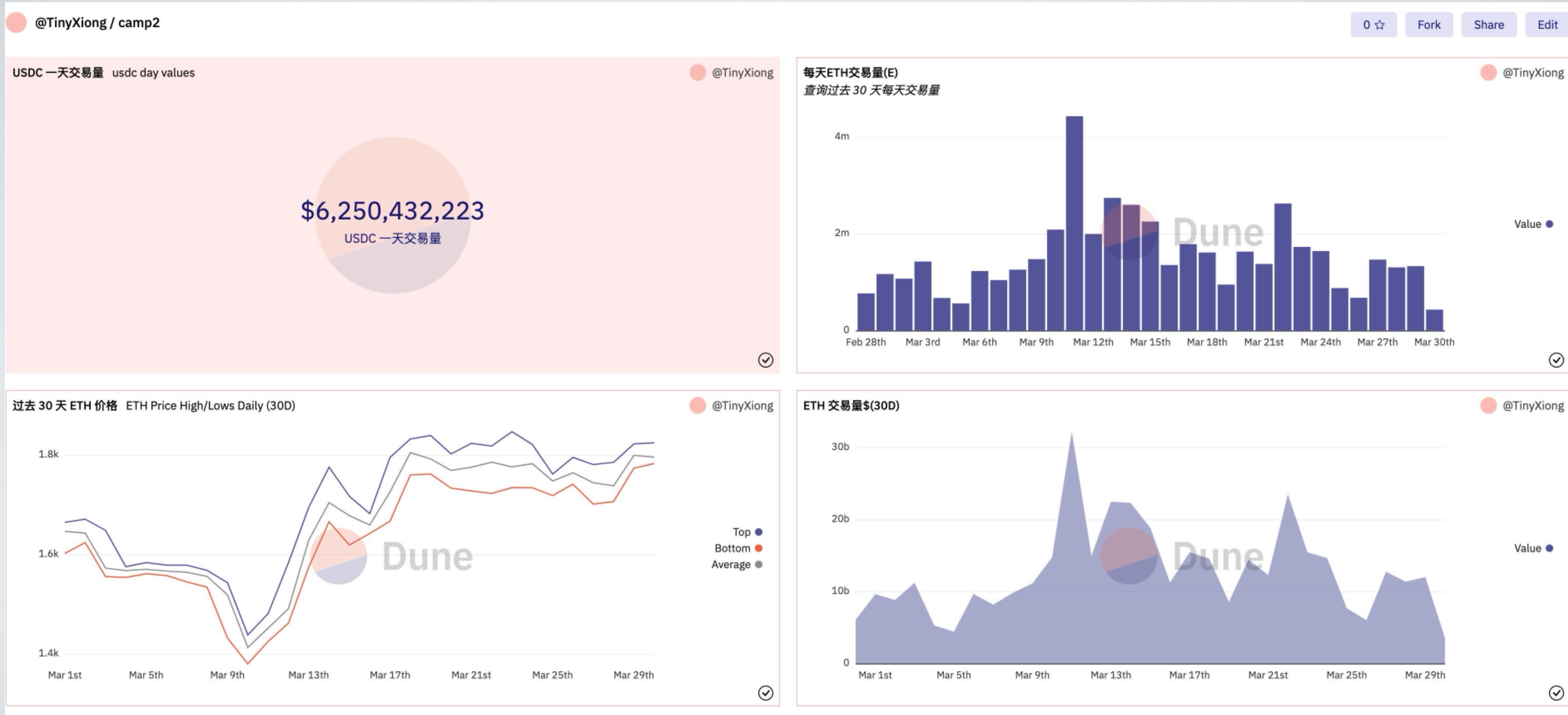


# DUNE 查询演练

- 查询以太坊上过去一天的 ETH 交易量 (ethereum.transactions)
- 查询以太坊上 USDC 过去一天的交易量 (erc20\_ethereum.evt\_Transfer)
- 查询以太坊上过去一周每天的 ETH 交易量 (sum value)
- 查询 以太坊 U 价格 过去一周的价格 (prices.usd 、 prices.usd\_daily )
- 查询以太坊上过去一周每天的 ETH 以 USD 计算的交易量
  - 先获取每天交易量和价格作为中间表，再求  $\text{sum}(\text{交易量} * \text{价格})$

[https://github.com/lbc-team/hello\\_thegraph/blob/main/dunesql\\_demo.sql](https://github.com/lbc-team/hello_thegraph/blob/main/dunesql_demo.sql)

# 查询仪表板展示





# 如何创建仪表盘

- 创建（或分叉）一些查询以获取一些区块链数据
- 使用这些查询来创建可视化效果
- 将可视化效果添加在一起以创建仪表板。

<https://dune.com/TinyXiong/camp2>

在 Dune 上，所有查询和数据集默认都是公开的，可复用，可查阅。

# 需求

- 为 PEPE 创建仪表盘（类似浏览器）：
  - 查看转账记录
  - 持币用户数数据：持币用户、持币余额
  - 总持币用户数、总发行量
  - 每天的活跃情况（交易笔数、交易数量、交易金额）

<https://dune.com/xilibi2003/pepe>

<https://etherscan.io/token/0x6982508145454ce325ddb47a25d4ec3d2311933>



# 如何写高效的查询

- 链上数量很大，避免读所有数据，
- 仅请求需要的列，减少查询引擎需要处理的数据量。
- 使用 limit 避免处理不必要的数据，避免使用 orderBy
- 使用基于非随机且可以排序的列的过滤条件（Dune 中，几乎所有表都是按时间和/或区块号分区的）
  - 如使用区块号\区块时间，而不是交易hash，当知道部署的区块高度时
  - 不要在过滤上使用函数，如 `date_trunc('day', block_time) > '2022-01-01'`
- 应该总是将较小的表连接到较大的表，内连接通常比外连接更快。

<https://docs.dune.com/query-engine/writing-efficient-queries>



# 可视化- 图表

- 默认情况下查询结果以表格形式呈现，但用户可以根据数据特点选择合适的图表，使数据更直观
- 折线图、柱状图、面积图、散点图、饼图等， **时间序列趋势**适合用折线图展示（如日交易量走势）， **类别占比对比**适合用柱状图或饼图
- SQL 查询中用 `date_trunc` 等函数生成不同粒度的时间序列



# Restful API 平台

- SIM API: 低延迟数据 API
  - <https://docs.sim.dune.com/evm/balances>
- Dune API : Execute Query / Get Execution Result
- Alchemy API
- QuickNode API
- Moralis
- GoldRush
- Helius (Solana)
- ...

[https://github.com/lbc-team/hello\\_thegraph/tree/main/restfulapi-demo](https://github.com/lbc-team/hello_thegraph/tree/main/restfulapi-demo)

# 作业：Dune 实践

- 创建 Azuki NFT 代币仪表盘，包括：
  - 持有者总数
  - 持有人名单（持有人及持有的NFT数量）
  - 最近一周每天在 OpenSea 的交易量（结合 `opensea.trades` / `nft.trades`）
- 提交可以公开访问的Dune看板链接

<https://decert.me/challenge/4087c3df-dfcd-4a3b-a51c-3120161aaa37>