Descrizione della Prova

In allegato troverete il file `43114_txs.csv.tar.gz` contenente 1 milione di transazioni effettuate sulla blockchain C-Chain tra il `4 ottobre 2024 alle 04:42:41` e il `10 ottobre 2024 alle 19:47:59`, specificamente tra i blocchi `51339620` e `51614312`.

Obiettivi

Importare le Transazioni
Importare queste transazioni in una base di dati appropriata. Assicurarsi che lo schema scelto sia ottimizzato per le query future e che garantisca un'alta velocità di accesso ai dati.

Processo di Inserimento in Real-Time (TypeScript)
Scrivere un processo in TypeScript che mantenga aggiornato il database inserendo le nuove transazioni iniettate nella C-Chain. Non è necessario importare i blocchi storici; il focus è solo sulle transazioni eseguite a partire dall'avvio del processo.

Processo di Esposizione Dati (TypeScript)
Scrivere un processo in TypeScript per esporre i dati memorizzati tramite API. Gli endpoint da implementare sono:

- Lista di transazioni: Restituire tutte le transazioni (opportunamente paginate) fatte o ricevute da un certo indirizzo, ordinate per blockNumber e transactionIndex.
- Conteggio transazioni: Fornire il numero di transazioni fatte o ricevute da un determinato indirizzo.
- Lista transazioni ordinate per valore: Restituire le transazioni ordinate per value (opportunamente paginate), cioè il quantitativo di $AVAX trasferito.

Note Aggiuntive

Ottimizzazione
Prestare particolare attenzione all'ottimizzazione delle query e dello schema del database per garantire risposte rapide agli endpoint. Considerando che la blockchain C-Chain attualmente conta oltre 500 milioni di transazioni, il sistema deve essere scalabile e garantire alte prestazioni anche con

una grande mole di dati.
Un singolo indirizzo potrebbe avere milioni di transazioni, quindi le query devono essere particolarmente ottimizzate per gestire tali situazioni senza degradare le prestazioni.

Hot Addresses per Testing

Forniamo una lista di indirizzi che hanno effettuato o ricevuto molte transazioni nel dataset allegato, utile per eventuali test di performance.

Indirizzi con più transazioni come mittenti

```
select "from", count() as txs from transactions group by "from" order by txs
desc limit 10;
```

```
┌──────────────────────────────────────────┬─────────┐
│                  from                    │   txs   │
│                varchar                   │  int64  │
├──────────────────────────────────────────┼─────────┤
│ 0x995BE1CA945174D5bA75410C1E658a41eB13a2FA │ 29325 │
│ 0xbD8679cf79137042214fA4239b02F4022208EE82 │ 14656 │
│ 0x9f8c163cBA728e99993ABe7495F06c0A3c8Ac8b9 │  9901 │
│ 0xCddc5d0Ebeb71a08ffF26909AA6c0d4e256b4fE1 │  9590 │
│ 0xffB3118124cdaEbD9095fA9a479895042018cac2 │  9405 │
│ 0x3BCE63C6C9ABf7A47f52c9A3a7950867700B0158 │  9316 │
│ 0xABa2D404C5C41da5964453A368aFF2604Ae80A14 │  9216 │
│ 0x6D8bE5cdf0d7DEE1f04E25FD70B001AE3B907824 │  9139 │
│ 0x2A44Ae6a71788FB62988bfE294e16063983BeC78 │  9040 │
│ 0x7E4aA755550152a522d9578621EA22eDAb204308 │  8675 │
└──────────────────────────────────────────┴─────────┘
```

Indirizzi con più transazioni come destinatari

```
select "to", count() as txs from transactions group by "to" order by txs desc
limit 10;
```

```
┌──────────────────────────────────────────┬─────────┐
│                    to                    │   txs   │
│                varchar                   │  int64  │
├──────────────────────────────────────────┼─────────┤
│ 0x9702230A8Ea53601f5cD2dc00fDBc13d4dF4A8c7 │ 161658 │
```

```
| 0x18556DA13313f3532c54711497A8FedAC273220E |  49145 |
| 0x069571BDA2f6Fbb2fEDcF18a7BbdA1014b485818 |  37794 |
| 0x802b65b5d9016621E66003aeD0b16615093f328b |  32556 |
| 0xB97EF9Ef8734C71904D8002F8b6Bc66Dd9c48a6E |  31087 |
| 0x7d2F62b6134e8B5676D17f5E988B3e48b4f50b70 |  23854 |
| 0x6203c968Ae2C15a562C53E0258FCB62e8139BD3E |  15447 |
| 0x2A375567f5E13F6bd74fDa7627Df3b1Af6BfA5a6 |  14661 |
| 0xef0cdae2FfEEeFA539a244a16b3f46ba75b8c810 |  11912 |
| 0x1a1ec25DC08e98e5E93F1104B5e5cdD298707d31 |  10737 |
└────────────────────────────────────────────┴────────┘
```

Link Utili

- C-Chain RPC: https://api.avax.network/ext/bc/C/rpc
- C-Chain JSON-RPC documentation: https://docs.infura.io/infura/networks/ethereum/json-rpc-methods
- https://viem.sh/