

02. The Onchain Stack: SQL, Spellbook, and Decoding UTXOs

“You can’t analyze what you don’t understand. The onchain stack starts with knowing your data.”

Every onchain analyst begins their journey not with a dashboard—but with a schema. To build meaningful insights, you need to understand how blockchain data is structured, what types of interactions it captures, and how to retrieve exactly what you need.

This article dives into the foundation of the onchain data stack. From Bitcoin’s UTXO model to Ethereum’s account model, from writing reusable queries in Dune’s Spellbook to designing your first modular dashboard—we’ll show you how real analysts read the chain.

First, Know the Models: UTXO vs. Account-Based

The two primary models in blockchain data are fundamentally different.

Bitcoin: UTXO (Unspent Transaction Output)

Bitcoin doesn’t track balances. Instead, every wallet’s “balance” is the sum of unspent outputs it controls. Each transaction consumes previous outputs and generates new ones—like handing in old bills to receive change.

Key tables on Dune:

- `bitcoin.outputs` : Where each BTC was sent
- `bitcoin.inputs` : Where each BTC came from (linked to previous outputs)

Ethereum: Account-Based

Ethereum uses a stateful model like a bank ledger. Wallets (EOAs) and smart contracts (CAs) have balances, and transactions mutate this shared state.

Key tables on Dune:

- `ethereum.transactions` : Every transaction on Ethereum
 - `ethereum.token_transfers` : ERC-20 token activity
 - `ethereum.logs` : Raw event logs emitted by contracts
-




Meet the Spellbook: Abstraction at Scale

Dune's **Spellbook** is a powerful open-source layer that lets you create reusable views across complex data.

Instead of repeating long subqueries or joining six tables every time, you can define a **spell** once—then use it like any regular table.

For example, rather than querying five different smart contracts for SpaceID registrations, you can define a spell called `spaceid_bnb_registrations` that abstracts this logic. Other users can then build on top of your work without copying and pasting raw SQL.

Benefits:

-  Better maintainability
-  Live updates for all dashboards when the spell is updated
-  A shared knowledge layer across the analyst community

Spellbook = open-source data infrastructure.

Example: Coin Days Destroyed (CDD) on Bitcoin

CDD is one of the most important behavioral indicators in Bitcoin analytics. It combines **value** and **time** by calculating how long coins have been dormant before they move.

The Formula:

Coin Days = BTC amount * Days since last moved

When coins move → Coin Days are "destroyed"

Large spikes in CDD signal movement from long-term holders—often considered “smart money.” These events often precede market turning points.

On Dune:

You can build this by:

- Fetching **bitcoin.outputs** (the creation of UTXOs)
- Linking to **bitcoin.inputs** (the spend of UTXOs)
- Calculating the time delta between them
- Multiplying that by the BTC amount

You’ve just built an economic time-weighted activity metric—with zero API dependencies.

Design Principles for Onchain Dashboards

Good dashboards don’t just visualize—they **answer questions**.

Here’s a framework:

1. What is the thing?

- Show the state: daily CDD, active addresses, volume, etc.

2. Why did it change?

- Decompose across dimensions: time of day, wallet type, transaction size.

Example: A spike in CDD might come from a **single whale**, or a **broad movement** of dormant wallets. Your dashboard should show both.

Tools You'll Use

Tool	Purpose
Dune Spellbook	Abstract repeated logic into reusable SQL views
DBT	Compiles and tests spells in the Spellbook repo
Git + GitHub	Collaborate on community-driven data pipelines

Want to contribute? Fork the [Dune Spellbook repo](#), write a new spell, and submit a PR. Your work could power dashboards used by thousands.

You Are Not Just Querying—You're Engineering Insights

This is what separates surface-level data pulls from true onchain analytics. When you know how to:

- Decipher UTXO flows
- Join event logs with contract interactions
- Build modular spells that others can depend on
- Understand both data *and* the systems that generate it

...you stop being a user of data and start becoming a builder of meaning.

Up Next

Now that we've covered the foundational models and infrastructure, it's time to get hands-on.

In the next article, we'll walk you through the Dune platform itself—how to write queries, use table explorers, and start building your first onchain dashboard.

Next: The Dune Platform — A Gateway to Onchain Transparency