

04. Understanding Tables — Ethereum, Bitcoin, NFTs, ERC4337, and More

If the blockchain is the raw ledger, and Dune is the window into it—then tables are the DNA of onchain analytics.

To do real work as an onchain data analyst, you need to know how blockchain data is structured, how Dune organizes it, and how to query the right fields.

This article walks through the anatomy of blockchain tables—covering Ethereum, Bitcoin, ERC4337 (Account Abstraction), NFTs, and more.

The EVM Data Model

Most blockchains supported by Dune are **EVM-compatible** (Ethereum Virtual Machine), meaning they use the same account-based structure:

- **EOAs (Externally Owned Accounts):** wallets controlled by private keys
- **Contract Accounts:** smart contracts that execute code when called

Every action on an EVM chain emits:

- A **transaction**
- One or more **events** (emitted from smart contracts)
- Changes to **state** (tracked via internal calls)

Dune decodes all of these into readable tables.

Core Ethereum Tables

Here are the most essential tables you'll use on Ethereum and other EVM chains:

ethereum.transactions

Every transaction sent on the network.

Key fields:

- `hash` : transaction hash
- `from` , `to` : sender and recipient
- `value` : ETH transferred
- `gas_used` , `gas_price` : cost of execution
- `block_time` : timestamp

ethereum.logs

Decoded event logs from smart contracts.

Use this to capture things like:

- Token transfers
- NFT mints
- DAO proposals
- DEX swaps

You'll often filter by `event_name` and `contract_address` .

`ethereum.token_transfers`

Normalized view of ERC20 transfers (built from `logs`).

Key fields:

- `token_address` : contract address of the token
- `from , to` : sender and receiver
- `amount` : token amount
- `block_time` : when it happened

This is useful for tracking token flow, whale movements, or airdrops.

`prices.usd`

Maps token prices over time for consistent USD calculations.

Join on:

- `contract_address`
 - `minute` (rounded timestamp)
-

NFT Tables

For NFTs, Dune often decodes each major collection or marketplace into its own schema.

Look for schemas like:

- `seaport_ethereum` (OpenSea)
- `blur_ethereum`

- `nft_ethereum`
- `erc721_transfers`

Track:

- `minted` , `sold` , `transferred` , `burned`
- Price, buyer, seller, royalty fees

Bitcoin Tables: UTXO Model

Unlike Ethereum's account model, Bitcoin uses a **UTXO** (Unspent Transaction Output) model.

That means there is no “account balance.” Instead, wallets hold a collection of unspent outputs.

Dune provides:

`bitcoin.inputs`

Each input spent in a BTC transaction.

Fields:

- `address` : the spender
 - `value` : amount of BTC
 - `block_time` : when the transaction occurred
-

bitcoin.outputs

Each output generated in a BTC transaction.

Fields:

- `address` : the receiver
- `value` : BTC received
- `block_time` : timestamp

Together, these two tables allow analysis of HODLing behavior, coin lifespan, and metrics like **Coin Days Destroyed (CDD)**.

ERC-4337 Tables (Account Abstraction)

ERC-4337 introduces a new way of transacting using smart contract wallets.

Dune decodes these contracts into a special schema— `erc4337` .

`erc4337_<chain>.EntryPoint_v0_6_evt_UserOperationEvent`

Each user operation submitted via a smart wallet.

Fields to know:

- `sender` : the smart wallet
- `paymaster` : who paid gas
- `actualGasCost` : gas used
- `success` : if the operation succeeded

Also explore:

- `AccountDeployed` events
 - `handleOps()` calls
 - `EntryPoint` contract interactions
-

Aggregated Spellbook Tables

Dune's community-built **Spellbook** offers aggregated, cleaned-up models across chains.

Instead of querying 9 versions of a table, use:

- `account_abstraction_erc4337.userops` : unified ERC-4337 activity
- `uniswap_v3.uniswap_v3_swaps` : all Uniswap V3 swaps across chains
- `nft.trades` : normalized NFT sales data

We'll explore Spellbook more later—just know these **save time** and **boost consistency**.

Best Practices

- Always filter by `block_time >` to avoid scanning full history
 - Use `date_trunc()` to group time series
 - Join with `prices.usd` to convert token amounts to USD
 - Use `LIMIT` when exploring new tables
 - Use `LOWER()` to normalize addresses when joining
 - Explore schema docs in the left-hand panel of Dune's editor
-

Where to Find Table Names

In the Dune query editor:

- Look to the left sidebar
- Browse by chain → schema → table
- Click a table to preview fields and structure

Or type `/ethereum.` to trigger autocomplete.

You Are What You Query

Every meaningful dashboard starts with understanding the structure beneath it.

In Web2, this might mean knowing the difference between GA4 and Mixpanel.

In Web3, it means understanding tables like:

- `ethereum.transactions`
- `erc20.token_transfers`
- `bitcoin.outputs`
- `erc4337.UserOperationEvent`
- `spellbook.models.uniswap_v3_swaps`

Once you master the table structure, the data opens up.

Next: 05. SQL Basics for Blockchain Analytics