

CrowdChain Entity Relationship Diagram (ERD)

Explanation

CrowdChain ERD

Overview

This Entity Relationship Diagram (ERD) represents the database structure for the CrowdChain application, a blockchain-based crowdfunding platform with milestone-based disbursement functionality. The database design balances on-chain and off-chain data storage to optimize for both decentralization and cost-efficiency.

On-Chain vs. Off-Chain Data Storage

Data Stored On-Chain (Blockchain)

- **Core Project Data:** Basic project information including ID, title, description, target amount, and status
- **Funding Transactions:** Records of contributions made by backers
- **Milestone Disbursements:** Actual fund transfers to project owners upon milestone completion
- **Smart Contract State:** Overall statistics like total projects, total backings, and total donations

Data Stored Off-Chain (Supabase)

- **Extended User Profiles:** Detailed user information beyond wallet addresses
- **Project Details:** Additional metadata, images, and categorization
- **Milestone Definitions and Tracking:** Detailed milestone information and completion status
- **Voting and Governance:** Backer votes on milestone completion
- **Comments and Updates:** Project communication and community engagement
- **Transaction References:** References to on-chain transactions for UI display and tracking

Entity Descriptions

Users

This entity stores user profile information beyond what's available on the blockchain.

- **id:** Primary key, unique identifier for each user
- **wallet_address:** User's blockchain wallet address (unique)

- **username:** User's chosen display name
- **email:** User's email address for notifications
- **avatar_url:** URL to user's profile image
- **bio:** Short user description
- **created_at:** Timestamp of account creation
- **updated_at:** Timestamp of last profile update

Projects

This entity stores project information, extending what's stored on the blockchain.

- **id:** Primary key, unique identifier for each project in Supabase
- **blockchain_id:** Reference to the project ID on the blockchain
- **owner_id:** Foreign key referencing the project creator in the Users table
- **title:** Project title
- **description:** Detailed project description
- **image_url:** URL to project cover image
- **target_amount:** Total funding goal
- **deadline:** Project funding deadline
- **status:** Current project status (draft, active, funded, completed, cancelled)
- **category:** Project category
- **tags:** Array of tags for improved searchability
- **created_at:** Timestamp of project creation
- **updated_at:** Timestamp of last project update

Milestones

This entity is central to the milestone-based disbursement feature, allowing projects to receive funding in stages.

- **id:** Primary key, unique identifier for each milestone
- **project_id:** Foreign key referencing the associated project
- **title:** Milestone title
- **description:** Detailed description of the milestone deliverables
- **amount:** Funding amount allocated to this milestone
- **due_date:** Expected completion date
- **status:** Current status (pending, in_progress, completed, approved, rejected)
- **completion_proof:** URL or text evidence of milestone completion
- **order_number:** Sequence number for milestone ordering
- **created_at:** Timestamp of milestone creation
- **updated_at:** Timestamp of last milestone update

Pledges

This entity tracks backer contributions, with references to both on-chain and off-chain data.

- **id**: Primary key, unique identifier for each pledge
- **backer__id**: Foreign key referencing the backer in the Users table
- **project__id**: Foreign key referencing the associated project
- **blockchain__id**: Reference to the backing ID on the blockchain
- **amount**: Contribution amount
- **status**: Current status (pending, confirmed, refunded)
- **created__at**: Timestamp of pledge creation
- **updated__at**: Timestamp of last pledge update

MilestoneDisbursements

This entity tracks the actual disbursement of funds when milestones are completed and approved.

- **id**: Primary key, unique identifier for each disbursement
- **milestone__id**: Foreign key referencing the associated milestone
- **tx_hash**: Transaction hash on the blockchain
- **amount**: Amount disbursed
- **status**: Current status (pending, completed, failed)
- **disbursement__date**: Date when funds were disbursed
- **created__at**: Timestamp of disbursement record creation
- **updated__at**: Timestamp of last disbursement record update

MilestoneVotes

This entity enables backers to vote on milestone completion, adding a governance layer to fund disbursement.

- **id**: Primary key, unique identifier for each vote
- **milestone__id**: Foreign key referencing the associated milestone
- **user__id**: Foreign key referencing the voter
- **vote**: Boolean indicating approval or rejection
- **comment**: Optional comment explaining the vote
- **created__at**: Timestamp of vote creation
- **updated__at**: Timestamp of last vote update

BlockchainTransactions

This entity tracks all blockchain transactions related to the application for reference and auditing.

- **id**: Primary key, unique identifier for each transaction record
- **user__id**: Foreign key referencing the associated user
- **tx_hash**: Transaction hash on the blockchain
- **tx_type**: Type of transaction (pledge, disbursement, refund)
- **amount**: Transaction amount
- **status**: Current status (pending, confirmed, failed)
- **created__at**: Timestamp of transaction record creation

- **updated_at**: Timestamp of last transaction record update

ProjectUpdates

This entity allows project creators to post updates about their projects.

- **id**: Primary key, unique identifier for each update
- **project_id**: Foreign key referencing the associated project
- **user_id**: Foreign key referencing the update author
- **title**: Update title
- **content**: Update content
- **created_at**: Timestamp of update creation
- **updated_at**: Timestamp of last update edit

Comments

This entity enables community discussion on projects.

- **id**: Primary key, unique identifier for each comment
- **project_id**: Foreign key referencing the associated project
- **user_id**: Foreign key referencing the comment author
- **content**: Comment content
- **created_at**: Timestamp of comment creation
- **updated_at**: Timestamp of last comment edit

Categories

This entity provides a way to categorize projects for better organization and discovery.

- **id**: Primary key, unique identifier for each category
- **name**: Category name
- **description**: Category description
- **created_at**: Timestamp of category creation
- **updated_at**: Timestamp of last category update

Relationships

1. **Users to Projects** (One-to-Many): A user can create multiple projects, but each project has only one owner.
2. **Users to Pledges** (One-to-Many): A user can make multiple pledges, but each pledge is associated with only one backer.
3. **Projects to Milestones** (One-to-Many): A project can have multiple milestones, but each milestone belongs to only one project.
4. **Projects to Pledges** (One-to-Many): A project can receive multiple pledges, but each pledge is associated with only one project.

5. **Milestones to MilestoneDisbursements** (One-to-Many): A milestone can have multiple disbursement attempts (in case of failures), but each disbursement is associated with only one milestone.
6. **Milestones to MilestoneVotes** (One-to-Many): A milestone can receive multiple votes from different backers, but each vote is associated with only one milestone.
7. **Users to MilestoneVotes** (One-to-Many): A user can vote on multiple milestones, but each vote is cast by only one user.
8. **Users to BlockchainTransactions** (One-to-Many): A user can be associated with multiple blockchain transactions, but each transaction record is associated with only one user.
9. **Projects to ProjectUpdates** (One-to-Many): A project can have multiple updates, but each update is associated with only one project.
10. **Projects to Comments** (One-to-Many): A project can have multiple comments, but each comment is associated with only one project.
11. **Users to ProjectUpdates** (One-to-Many): A user can create multiple project updates, but each update is created by only one user.
12. **Users to Comments** (One-to-Many): A user can post multiple comments, but each comment is posted by only one user.

Milestone-Based Disbursement Implementation

The milestone-based disbursement functionality is implemented through several interconnected entities:

1. **Milestones**: Define the deliverables, funding amounts, and timelines for each project phase.
2. **MilestoneVotes**: Enable backers to participate in governance by voting on milestone completion.
3. **MilestoneDisbursements**: Track the actual fund transfers that occur when milestones are approved.

This system provides several benefits:

- **Risk Reduction**: Backers' funds are released incrementally as the project demonstrates progress.
- **Accountability**: Project creators must deliver on promises before receiving full funding.
- **Transparency**: All stakeholders can track project progress through clearly defined milestones.
- **Governance**: Backers have a say in whether milestones have been satisfactorily completed.

The smart contract would need to be modified to support this functionality, with functions for: - Creating milestones - Voting on milestone completion - Disbursing funds upon milestone approval - Handling refunds if projects fail to meet milestones

Conclusion

This database design balances on-chain and off-chain data storage to create an efficient, scalable crowdfunding platform with enhanced security and accountability through milestone-based disbursement. The Supabase implementation complements the blockchain data, providing rich features while minimizing gas costs and blockchain bloat.