

Database Tables and Relationships

1. Users

Stores information about the app's users.

Column Name	Data Type	Constraints
user_id	UUID	Primary Key
name	VARCHAR(255)	Not Null
email	VARCHAR(255)	Unique, Not Null
password_hash	VARCHAR(255)	Not Null
location	VARCHAR(255)	Nullable
points	INT	Default: 0
created_at	TIMESTAMP	Default: Current Timestamp
updated_at	TIMESTAMP	Auto-update on modification

2. Polls

Contains details about polls created by businesses or administrators.

Column Name	Data Type	Constraints
poll_id	UUID	Primary Key
title	VARCHAR(255)	Not Null
description	TEXT	Nullable
creator_id	UUID	Foreign Key (user_id)
is_active	BOOLEAN	Default: True
created_at	TIMESTAMP	Default: Current Timestamp

expires_at	TIMESTAMP	Nullable
------------	-----------	----------

3. Options

Stores the answer options for each poll.

Column Name	Data Type	Constraints
option_id	UUID	Primary Key
poll_id	UUID	Foreign Key (poll_id)
option_text	VARCHAR(255)	Not Null
vote_count	INT	Default: 0

4. Votes

Tracks user votes for polls.

Column Name	Data Type	Constraints
vote_id	UUID	Primary Key
poll_id	UUID	Foreign Key (poll_id)
option_id	UUID	Foreign Key (option_id)
user_id	UUID	Foreign Key (user_id)
created_at	TIMESTAMP	Default: Current Timestamp

5. Rewards

Tracks rewards that users can earn and redeem.

Column Name	Data Type	Constraints
reward_id	UUID	Primary Key

reward_name	VARCHAR(255)	Not Null
points_required	INT	Not Null
created_at	TIMESTAMP	Default: Current Timestamp

6. User_Rewards

Tracks which rewards users have redeemed.

Column Name	Data Type	Constraints
user_reward_id	UUID	Primary Key
user_id	UUID	Foreign Key (user_id)
reward_id	UUID	Foreign Key (reward_id)
redeemed_at	TIMESTAMP	Default: Current Timestamp

7. Analytics

Stores aggregated data for polls to support analytics.

Column Name	Data Type	Constraints
poll_id	UUID	Foreign Key (poll_id)
total_votes	INT	Default: 0
unique_voters	INT	Default: 0
last_updated	TIMESTAMP	Auto-update on modification

Relationships

1. **Users and Polls:** A user (creator) can create many polls (**1:N** relationship).
 2. **Polls and Options:** Each poll has multiple options for users to vote on (**1:N** relationship).
 3. **Users and Votes:** A user can vote on multiple polls, and each vote is tied to a specific option (**M:N** relationship using **Votes** table).
 4. **Users and Rewards:** Users can redeem multiple rewards (**M:N** relationship using **User_Rewards** table).
 5. **Polls and Analytics:** Each poll can have aggregated analytics data (**1:1** relationship).
-

Scalability Considerations

- Use **UUIDs** for primary keys to ensure globally unique identifiers across distributed systems.
- Implement indexes on frequently queried fields like **user_id**, **poll_id**, and **option_id** for faster lookups.
- Consider partitioning the **Votes** table for high-volume data, especially if you expect millions of votes.

This schema provides a solid foundation for your MVP, ensuring flexibility for future feature expansions. Let me know if you'd like further refinements!

8. Surveys

Contains information about surveys created by businesses or administrators, which may have multiple questions.

Column Name	Data Type	Constraints
survey_id	UUID	Primary Key
title	VARCHAR(255)	Not Null
description	TEXT	Nullable
creator_id	UUID	Foreign Key (user_id)
is_active	BOOLEAN	Default: True
created_at	TIMESTAMP	Default: Current Timestamp
expires_at	TIMESTAMP	Nullable

9. Survey_Questions

Stores individual questions for each survey.

Column Name	Data Type	Constraints
question_id	UUID	Primary Key
survey_id	UUID	Foreign Key (survey_id)
question_text	TEXT	Not Null
question_type	VARCHAR(50)	Enum: 'text', 'multiple_choice', 'rating'

10. Survey_Responses

Tracks user responses to survey questions.

Column Name	Data Type	Constraints
response_id	UUID	Primary Key
survey_id	UUID	Foreign Key (survey_id)
question_id	UUID	Foreign Key (question_id)
user_id	UUID	Foreign Key (user_id)
response_value	TEXT	Stores the user's response
created_at	TIMESTAMP	Default: Current Timestamp

Relationships for Surveys

1. **Surveys and Survey_Questions:** Each survey can have multiple questions (1:N relationship).
 2. **Survey_Questions and Survey_Responses:** Each question can have multiple responses from users (1:N relationship).
 3. **Users and Surveys:** A user can respond to multiple surveys, and each response links back to a specific survey and question (M:N relationship via **Survey_Responses**).
-

Scalability Considerations for Surveys

- For **Survey_Responses**, consider using a partitioned table if responses grow significantly.
- Store **question_type** in **Survey_Questions** to support different input types (e.g., text, multiple-choice, ratings).
- Use indexes on **survey_id** and **question_id** for fast response aggregation.