Supabase Project Report: kszrzybbmdzfsouztknz

Overview

This report provides a detailed overview of the database structure and data organization for the Supabase project "kszrzybbmdzfsouztknz". The project tracks personality assessments and game progressions, storing data in a structured format for analytics and monitoring.

Database Structure

1. Scores Table

- **Columns**:
  - `id`: UUID, unique identifier for each score record.
  - `score`: Double precision, nullable field to store the score.
  - `updated_at`: Timestamp, nullable, default value is set to the current time.
- **Purpose**: This table stores individual score records that can be associated with tasks or evaluations.

2. Personality Assessment Tables

- **Assessment Framework**:
  - **ID**: `3ac68f05-2ea9-4223-b139-d88373859379`
  - **Type**: Rating scale (1-5)
  - **Title**: "Big Five"
  - **Description**: "Big 5 Personality at Work"
- **Personality Dimensions**:
  - Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism
- **Data Storage**:
  - **questions**: Contains 50 questions categorized by personality trait with feature labels

(e.g., EXT1, AGR2).
    - **answers**: Stores individual responses with rating (1-5), links to question_id, user_id, assessment_id.
    - **evaluations**: Stores computed personality scores as JSONB objects with normalized scores (0-1) for each trait.

3. Game Progression Table

- **Table Name**: `game_info`
- **Types of Games**: Maze, Plane, Turtle
- **Game Progression Tracking**:
    - **Fields**:
        - `id`: UUID, primary key.
        - `profile_id`: UUID, foreign key linking to user profiles.
        - `game_id`: Text, game type identifier.
        - `levels_completed`: Array of booleans, tracks completed levels.
        - `durations`: Array of integers, stores level completion times.
        - `onboarding_completed`: Boolean, tracks onboarding status.

Sample Data Generation Plan

To generate synthetic data that mimics the structure of the real data, consider the following plan:

1. Scores Table

- Generate UUIDs for `id`.
- Randomly assign `score` values using a uniform distribution over a range that resembles typical scores.
- Set `updated_at` to random timestamps within a realistic range of dates.

2. Personality Assessment

- **Questions Table**: Use a fixed set of categorical questions.

- **Answers Table**: Assign `rating` values randomly (1-5).
- **Evaluations Table**:
  - Create random trait scores stored as JSONB with values ranging between 0 and 1.
  - Ensure realistic distribution (e.g., normal distribution centered around 0.5).

3. Game Progression

- **Levels Completed**: Use a binomial distribution to simulate partial progress (some levels completed, others not).
- **Durations**: Create durations based on a normal distribution to reflect likely completion times.
- **Onboarding Completed**: Use a uniform distribution to randomly mark onboarding as completed or not.

This strategy ensures that synthetic data closely resembles real-world data without exposing sensitive or proprietary information.