```sql
-- WARNING: This schema is for context only and is not meant to be run.
-- Table order and constraints may not be valid for execution.

CREATE TABLE public.budget_alerts (
  id uuid NOT NULL DEFAULT gen_random_uuid(),
  user_id uuid NOT NULL,
  budget_id uuid,
  alert_type character varying NOT NULL,
  category character varying,
  threshold_percentage integer DEFAULT 80,
  amount numeric,
  message text,
  is_active boolean DEFAULT true,
  triggered_at timestamp with time zone,
  created_at timestamp with time zone DEFAULT now(),
  CONSTRAINT budget_alerts_pkey PRIMARY KEY (id),
  CONSTRAINT budget_alerts_user_id_fkey FOREIGN KEY (user_id) REFERENCES auth.users(id),
  CONSTRAINT budget_alerts_budget_id_fkey FOREIGN KEY (budget_id) REFERENCES public.budgets(id)
);
CREATE TABLE public.budget_categories (
  id uuid NOT NULL DEFAULT gen_random_uuid(),
  budget_id uuid NOT NULL,
  category_name character varying NOT NULL,
  allocated_amount numeric NOT NULL DEFAULT 0,
  spent_amount numeric NOT NULL DEFAULT 0,
```

```sql
  created_at timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  CONSTRAINT budget_categories_pkey PRIMARY KEY (id),

  CONSTRAINT budget_categories_budget_id_fkey FOREIGN KEY (budget_id) REFERENCES
public.budgets(id)

);

CREATE TABLE public.budgets (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL UNIQUE,

  total_budget numeric NOT NULL DEFAULT 0,

  currency character varying NOT NULL DEFAULT 'PHP'::character varying,

  budget_period character varying NOT NULL DEFAULT 'monthly'::character varying,

  created_at timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  savings_goal numeric NOT NULL DEFAULT '0'::numeric,

  monthly numeric DEFAULT 0,

  weekly numeric DEFAULT 0,

  CONSTRAINT budgets_pkey PRIMARY KEY (id),

  CONSTRAINT budgets_user_id_fkey FOREIGN KEY (user_id) REFERENCES auth.users(id)

);

CREATE TABLE public.chat_messages (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  session_id uuid NOT NULL,

  user_id uuid NOT NULL,

  message_text text NOT NULL,

  is_bot_message boolean DEFAULT false,
```

```sql
  message_data jsonb DEFAULT '{}'::jsonb,

  timestamp timestamp with time zone NOT NULL DEFAULT timezone('utc'::text, now()),

  intent character varying,

  confidence_score numeric,

  response_time_ms integer,

  message_metadata jsonb DEFAULT '{}'::jsonb,

  CONSTRAINT chat_messages_pkey PRIMARY KEY (id),

  CONSTRAINT chat_messages_session_id_fkey FOREIGN KEY (session_id) REFERENCES
public.chat_sessions(id),

  CONSTRAINT chat_messages_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);
CREATE TABLE public.chat_sessions (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL,

  session_name character varying,

  created_at timestamp with time zone NOT NULL DEFAULT timezone('utc'::text, now()),

  updated_at timestamp with time zone NOT NULL DEFAULT timezone('utc'::text, now()),

  is_active boolean DEFAULT true,

  session_metadata jsonb DEFAULT '{}'::jsonb,

  CONSTRAINT chat_sessions_pkey PRIMARY KEY (id),

  CONSTRAINT chat_sessions_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);
CREATE TABLE public.chatbot_interactions (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL,
```

```sql
  user_input text NOT NULL,

  intent character varying,

  extracted_data jsonb,

  chatbot_response text,

  success boolean DEFAULT true,

  error_message text,

  processing_time_ms integer,

  created_at timestamp with time zone DEFAULT now(),

  CONSTRAINT chatbot_interactions_pkey PRIMARY KEY (id),

  CONSTRAINT chatbot_interactions_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);
CREATE TABLE public.expenses (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL,

  amount numeric NOT NULL,

  category character varying NOT NULL,

  note text,

  date timestamp with time zone DEFAULT now(),

  created_at timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  created_via text DEFAULT 'user'::text CHECK (created_via = ANY (ARRAY['user'::text,
'chatbot'::text])),

  confidence_score numeric CHECK (confidence_score >= 0::numeric AND
confidence_score <= 1::numeric),

  needs_review boolean DEFAULT false,

  natural_language_input text,
```

```sql
  CONSTRAINT expenses_pkey PRIMARY KEY (id),

  CONSTRAINT expenses_user_id_fkey FOREIGN KEY (user_id) REFERENCES auth.users(id)
);
CREATE TABLE public.friends (
 id uuid NOT NULL DEFAULT gen_random_uuid(),

 user_id uuid NOT NULL,

 friend_id uuid NOT NULL,

 status character varying DEFAULT 'pending'::character varying CHECK (status::text = ANY
(ARRAY['pending'::character varying::text, 'accepted'::character varying::text,
'blocked'::character varying::text, 'declined'::character varying::text])),

 requested_by uuid NOT NULL,

 requested_at timestamp with time zone DEFAULT now(),

 accepted_at timestamp with time zone,

 updated_at timestamp with time zone DEFAULT now(),

 CONSTRAINT friends_pkey PRIMARY KEY (id),

 CONSTRAINT friends_user_id_fkey FOREIGN KEY (user_id) REFERENCES auth.users(id),

 CONSTRAINT friends_friend_id_fkey FOREIGN KEY (friend_id) REFERENCES
auth.users(id),

 CONSTRAINT friends_requested_by_fkey FOREIGN KEY (requested_by) REFERENCES
auth.users(id)
);
CREATE TABLE public.gamified_savings_goals (
 id uuid NOT NULL DEFAULT gen_random_uuid(),

 user_id uuid NOT NULL,

 level integer NOT NULL CHECK (level >= 1 AND level <= 4),

 target_amount numeric NOT NULL,

 current_amount numeric DEFAULT 0.00,
```

```sql
  daily_target numeric NOT NULL,

  timeline_days integer NOT NULL,

  start_date timestamp with time zone DEFAULT now(),

  end_date timestamp with time zone NOT NULL,

  is_active boolean DEFAULT true,

  is_completed boolean DEFAULT false,

  completed_at timestamp with time zone,

  created_at timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  CONSTRAINT gamified_savings_goals_pkey PRIMARY KEY (id),

  CONSTRAINT gamified_savings_goals_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);

CREATE TABLE public.leaderboard_achievements (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL,

  achievement_type character varying NOT NULL,

  achievement_level integer DEFAULT 1,

  earned_at timestamp with time zone DEFAULT now(),

  criteria_value numeric,

  CONSTRAINT leaderboard_achievements_pkey PRIMARY KEY (id),

  CONSTRAINT leaderboard_achievements_user_id_fkey FOREIGN KEY (user_id)
REFERENCES auth.users(id)

);

CREATE TABLE public.leaderboards (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL UNIQUE,
```

```sql
  username character varying NOT NULL,

  profile_image_url text,

  total_saved numeric DEFAULT 0.00,

  highest_level integer DEFAULT 1,

  goals_completed integer DEFAULT 0,

  current_streak integer DEFAULT 0,

  longest_streak integer DEFAULT 0,

  total_transactions integer DEFAULT 0,

  avg_daily_savings numeric DEFAULT 0.00,

  rank_score numeric DEFAULT 0.00,

  last_activity timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  created_at timestamp with time zone DEFAULT now(),

  CONSTRAINT leaderboards_pkey PRIMARY KEY (id),

  CONSTRAINT leaderboards_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);
CREATE TABLE public.notes (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL,

  title text NOT NULL,

  content text,

  created_at timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  CONSTRAINT notes_pkey PRIMARY KEY (id),

  CONSTRAINT notes_user_id_fkey FOREIGN KEY (user_id) REFERENCES auth.users(id)
```

```sql
);

CREATE TABLE public.password_reset_codes (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  email text NOT NULL,

  code text NOT NULL,

  expires_at timestamp with time zone NOT NULL,

  used boolean DEFAULT false,

  created_at timestamp with time zone DEFAULT now(),

  CONSTRAINT password_reset_codes_pkey PRIMARY KEY (id)

);

CREATE TABLE public.profiles (

  id uuid NOT NULL,

  full_name text,

  email text UNIQUE,

  created_at timestamp with time zone NOT NULL DEFAULT now(),

  updated_at timestamp with time zone NOT NULL DEFAULT now(),

  onboarding_completed boolean DEFAULT false,

  username character varying UNIQUE CHECK (username IS NULL OR
length(username::text) >= 3 AND length(username::text) <= 30 AND username::text ~ '^[a-
zA-Z0-9_]+$'::text),

  user_type character varying CHECK (user_type IS NULL OR (user_type::text = ANY
(ARRAY['student'::character varying, 'employee'::character varying]::text[]))),

  CONSTRAINT profiles_pkey PRIMARY KEY (id),

  CONSTRAINT profiles_id_fkey FOREIGN KEY (id) REFERENCES auth.users(id)

);

CREATE TABLE public.savings_goal_transactions (

  id uuid NOT NULL DEFAULT gen_random_uuid(),
```

```sql
  user_id uuid NOT NULL,

  goal_id uuid NOT NULL,

  amount numeric NOT NULL,

  note text,

  transaction_date timestamp with time zone DEFAULT now(),

  created_at timestamp with time zone DEFAULT now(),

  CONSTRAINT savings_goal_transactions_pkey PRIMARY KEY (id),

  CONSTRAINT savings_goal_transactions_user_id_fkey FOREIGN KEY (user_id)
REFERENCES auth.users(id)

);

CREATE TABLE public.savings_goals (

  goal_id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL,

  level integer NOT NULL CHECK (level >= 1 AND level <= 4),

  target_amount numeric NOT NULL CHECK (target_amount > 0::numeric),

  current_amount numeric DEFAULT 0 CHECK (current_amount >= 0::numeric),

  daily_target numeric NOT NULL CHECK (daily_target > 0::numeric),

  start_date date NOT NULL DEFAULT CURRENT_DATE,

  end_date date NOT NULL,

  status character varying DEFAULT 'active'::character varying CHECK (status::text = ANY
(ARRAY['active'::character varying, 'completed'::character varying, 'failed'::character
varying]::text[])),

  created_at timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  completed_at timestamp with time zone,

  CONSTRAINT savings_goals_pkey PRIMARY KEY (goal_id),
```

```sql
  CONSTRAINT savings_goals_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)
);
CREATE TABLE public.savings_level_configs (
  id integer NOT NULL DEFAULT nextval('savings_level_configs_id_seq'::regclass),
  level integer NOT NULL UNIQUE CHECK (level >= 1 AND level <= 4),
  title character varying NOT NULL,
  description text,
  target_amount numeric NOT NULL,
  max_days integer NOT NULL,
  icon character varying,
  color character varying,
  created_at timestamp with time zone DEFAULT now(),
  CONSTRAINT savings_level_configs_pkey PRIMARY KEY (id)
);
CREATE TABLE public.savings_transactions (
  transaction_id uuid NOT NULL DEFAULT gen_random_uuid(),
  goal_id uuid,
  user_id uuid NOT NULL,
  amount numeric NOT NULL CHECK (amount > 0::numeric),
  transaction_date date NOT NULL DEFAULT CURRENT_DATE,
  notes text,
  created_at timestamp with time zone DEFAULT now(),
  id uuid DEFAULT gen_random_uuid(),
  transaction_type character varying DEFAULT 'deposit'::character varying CHECK
(transaction_type::text = ANY (ARRAY['deposit'::character varying, 'withdrawal'::character
varying]::text[])),
```

description text,

CONSTRAINT savings_transactions_pkey PRIMARY KEY (transaction_id),

CONSTRAINT savings_transactions_user_id_fkey FOREIGN KEY (user_id) REFERENCES auth.users(id),

CONSTRAINT savings_transactions_goal_id_fkey FOREIGN KEY (goal_id) REFERENCES public.gamified_savings_goals(id)

);

CREATE TABLE public.transactions (

 id uuid NOT NULL DEFAULT gen_random_uuid(),

 user_id uuid NOT NULL,

 budget_id uuid,

 amount numeric NOT NULL,

 category character varying NOT NULL,

 description text,

 transaction_date date NOT NULL DEFAULT CURRENT_DATE,

 created_via character varying NOT NULL DEFAULT 'chatbot'::character varying,

 natural_language_input text,

 confidence_score numeric,

 needs_review boolean DEFAULT false,

 created_at timestamp with time zone DEFAULT now(),

 updated_at timestamp with time zone DEFAULT now(),

 CONSTRAINT transactions_pkey PRIMARY KEY (id),

 CONSTRAINT transactions_user_id_fkey FOREIGN KEY (user_id) REFERENCES auth.users(id)

);

CREATE TABLE public.user_achievements (

 id uuid NOT NULL DEFAULT gen_random_uuid(),

```sql
  user_id uuid NOT NULL,

  achievement_id character varying NOT NULL,

  achievement_title character varying NOT NULL,

  achievement_description text,

  points integer DEFAULT 0,

  unlocked_at timestamp with time zone DEFAULT now(),

  CONSTRAINT user_achievements_pkey PRIMARY KEY (id),

  CONSTRAINT user_achievements_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);

CREATE TABLE public.user_chat_preferences (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL UNIQUE,

  enable_history boolean DEFAULT true,

  max_sessions integer DEFAULT 50,

  max_messages_per_session integer DEFAULT 500,

  auto_delete_after_days integer DEFAULT 90,

  context_memory_enabled boolean DEFAULT true,

  created_at timestamp with time zone NOT NULL DEFAULT timezone('utc'::text, now()),

  updated_at timestamp with time zone NOT NULL DEFAULT timezone('utc'::text, now()),

  CONSTRAINT user_chat_preferences_pkey PRIMARY KEY (id),

  CONSTRAINT user_chat_preferences_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);

CREATE TABLE public.user_favorite_tips (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL,
```

```sql
  tip_id character varying NOT NULL,

  favorited_at timestamp with time zone DEFAULT now(),

  CONSTRAINT user_favorite_tips_pkey PRIMARY KEY (id),

  CONSTRAINT user_favorite_tips_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);

CREATE TABLE public.user_learning_progress (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL,

  module_id character varying NOT NULL,

  lesson_id character varying NOT NULL,

  completed boolean DEFAULT false,

  score integer,

  completed_at timestamp with time zone,

  created_at timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  CONSTRAINT user_learning_progress_pkey PRIMARY KEY (id),

  CONSTRAINT user_learning_progress_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);

CREATE TABLE public.user_learning_stats (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL UNIQUE,

  total_lessons_completed integer DEFAULT 0,

  total_quizzes_taken integer DEFAULT 0,

  total_tips_favorited integer DEFAULT 0,

  average_quiz_score numeric DEFAULT 0,
```

```sql
  learning_streak_days integer DEFAULT 0,

  last_activity_date date,

  created_at timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  CONSTRAINT user_learning_stats_pkey PRIMARY KEY (id),

  CONSTRAINT user_learning_stats_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);

CREATE TABLE public.user_levels (

  user_id uuid NOT NULL,

  current_level integer DEFAULT 1 CHECK (current_level >= 1 AND current_level <= 4),

  total_saved numeric DEFAULT 0 CHECK (total_saved >= 0::numeric),

  goals_completed integer DEFAULT 0 CHECK (goals_completed >= 0),

  streak_days integer DEFAULT 0 CHECK (streak_days >= 0),

  last_save_date date,

  created_at timestamp with time zone DEFAULT now(),

  updated_at timestamp with time zone DEFAULT now(),

  CONSTRAINT user_levels_pkey PRIMARY KEY (user_id),

  CONSTRAINT user_levels_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);

CREATE TABLE public.user_quiz_results (

  id uuid NOT NULL DEFAULT gen_random_uuid(),

  user_id uuid NOT NULL,

  quiz_id character varying NOT NULL,

  lesson_id character varying,

  tip_id character varying,
```

```sql
  selected_answer integer NOT NULL,

  correct_answer integer NOT NULL,

  is_correct boolean NOT NULL,

  attempts integer DEFAULT 1,

  first_attempt_correct boolean NOT NULL,

  completed_at timestamp with time zone DEFAULT now(),

  created_at timestamp with time zone DEFAULT now(),

  CONSTRAINT user_quiz_results_pkey PRIMARY KEY (id),

  CONSTRAINT user_quiz_results_user_id_fkey FOREIGN KEY (user_id) REFERENCES
auth.users(id)

);
```