Database Management Assignment – PixaBeam

Supabase Database Design & Implementation

Name - Deepti Ranjan Das **Date -** 20/08/2025

Part 1 - Database Design

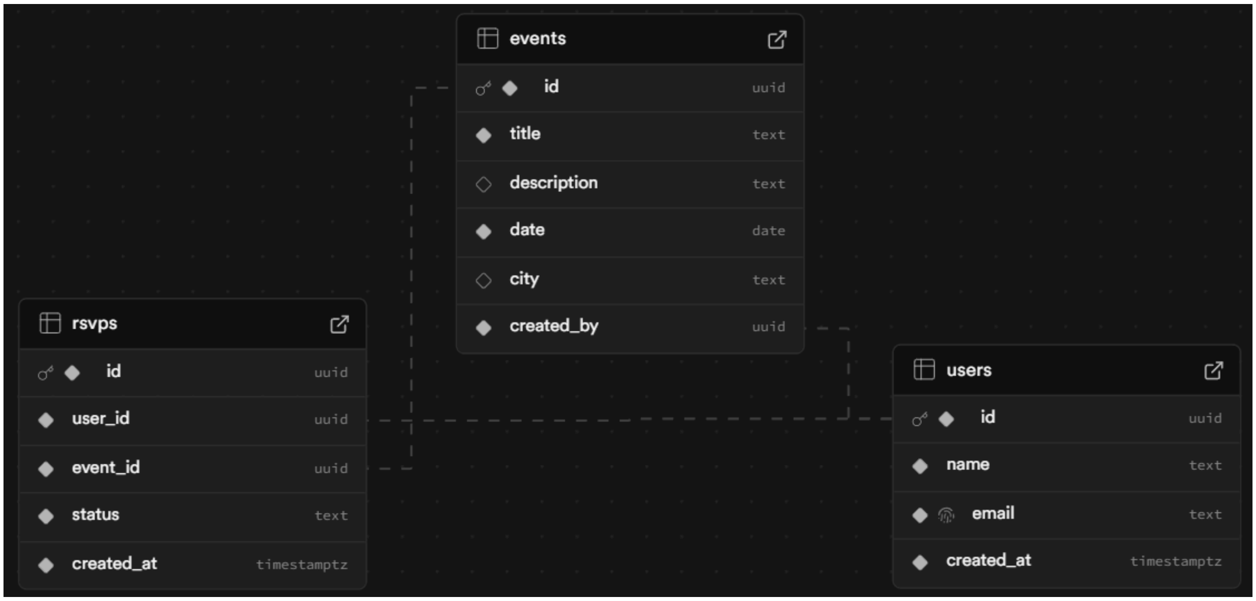
This assignment demonstrates my ability to design and implement a relational database in **Supabase**. The project involves creating tables for `Users`, `Events`, and `RSVPs`, ensuring referential integrity with primary and foreign keys, and inserting sample data. Additionally, I explain my design choices and provide supporting screenshots.

SQL Dump File:

```sql
-- Enable UUID generator
CREATE EXTENSION IF NOT EXISTS "pgcrypto";

-- Tables
CREATE TABLE IF NOT EXISTS public.users (
  id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
  name text NOT NULL,
  email text NOT NULL UNIQUE,
  created_at timestamptz NOT NULL DEFAULT now()
);

CREATE TABLE IF NOT EXISTS public.events (
  id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
  title text NOT NULL,
  description text,
  date date NOT NULL,
  city text,
  created_by uuid NOT NULL REFERENCES public.users(id) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS public.rsvps (
  id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id uuid NOT NULL REFERENCES public.users(id) ON DELETE CASCADE,
  event_id uuid NOT NULL REFERENCES public.events(id) ON DELETE CASCADE,
  status text NOT NULL,
  created_at timestamptz NOT NULL DEFAULT now(),
  CONSTRAINT rsvps_status_check CHECK (status IN ('Yes','No','Maybe')),
  CONSTRAINT rsvps_unique_user_event UNIQUE (user_id, event_id)
);
```

```
-- Helpful indexes
CREATE INDEX IF NOT EXISTS idx_events_date  ON public.events(date);
CREATE INDEX IF NOT EXISTS idx_events_city  ON public.events(city);
CREATE INDEX IF NOT EXISTS idx_rsvps_user   ON public.rsvps(user_id);
CREATE INDEX IF NOT EXISTS idx_rsvps_event  ON public.rsvps(event_id);
```

ER Diagram Screenshot:
Entity-Relationship diagram generated in Supabase, showing table relationships and constraints.

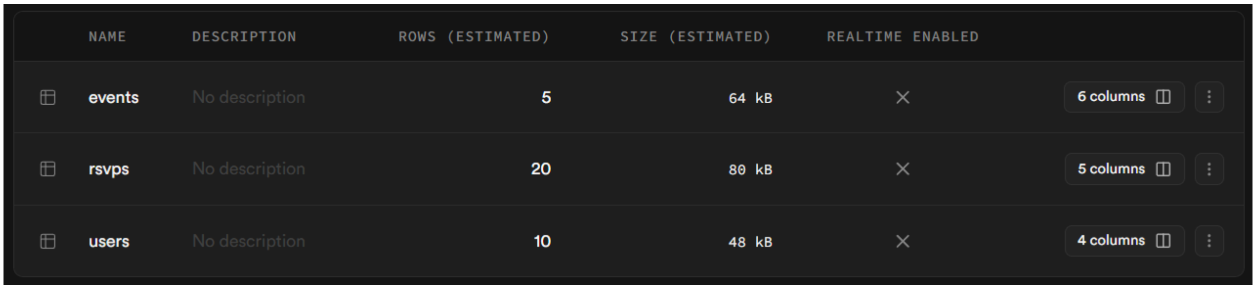


events	
id	uuid
title	text
description	text
date	date
city	text
created_by	uuid

rsvps	
id	uuid
user_id	uuid
event_id	uuid
status	text
created_at	timestamptz

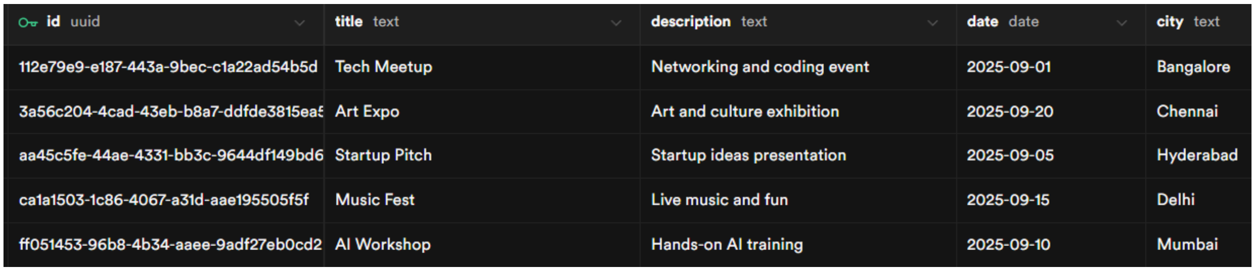
users	
id	uuid
name	text
email	text
created_at	timestamptz

Database Screenshots:



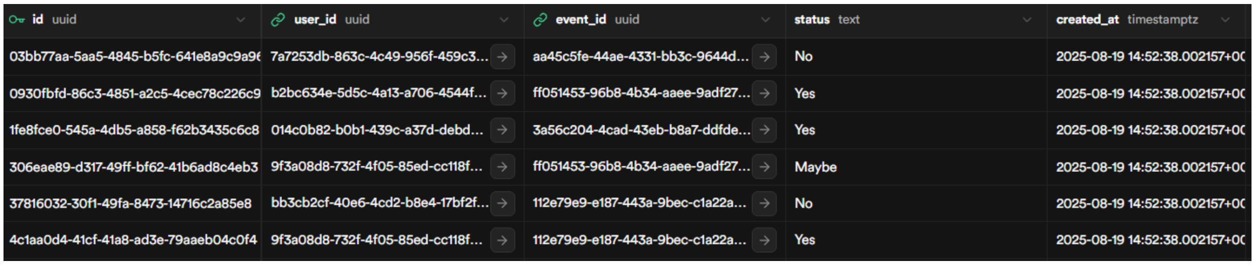
NAME	DESCRIPTION	ROWS (ESTIMATED)	SIZE (ESTIMATED)	REALTIME ENABLED	
events	No description	5	64 kB	✕	6 columns
rsvps	No description	20	80 kB	✕	5 columns
users	No description	10	48 kB	✕	4 columns

Events:



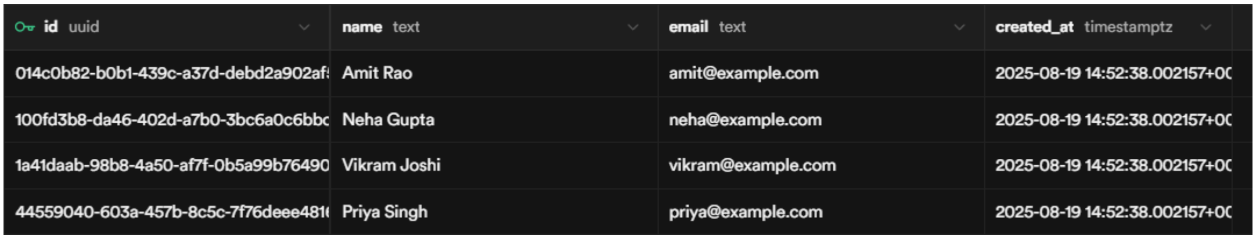
id uuid	title text	description text	date date	city text
112e79e9-e187-443a-9bec-c1a22ad54b5d	Tech Meetup	Networking and coding event	2025-09-01	Bangalore
3a56c204-4cad-43eb-b8a7-ddfde3815ea5	Art Expo	Art and culture exhibition	2025-09-20	Chennai
aa45c5fe-44ae-4331-bb3c-9644df149bd6	Startup Pitch	Startup ideas presentation	2025-09-05	Hyderabad
ca1a1503-1c86-4067-a31d-aae195505f5f	Music Fest	Live music and fun	2025-09-15	Delhi
ff051453-96b8-4b34-aaee-9adf27eb0cd2	AI Workshop	Hands-on AI training	2025-09-10	Mumbai

RSVPs:



id uuid	user_id uuid	event_id uuid	status text	created_at timestamptz
03bb77aa-5aa5-4845-b5fc-641e8a9c9a9	7a7253db-863c-4c49-956f-459c3...	aa45c5fe-44ae-4331-bb3c-9644d...	No	2025-08-19 14:52:38.002157+00
0930fbfd-86c3-4851-a2c5-4cec78c226c9	b2bc634e-5d5c-4a13-a706-4544f...	ff051453-96b8-4b34-aaee-9adf27...	Yes	2025-08-19 14:52:38.002157+00
1fe8fce0-545a-4db5-a858-f62b3435c6c8	014c0b82-b0b1-439c-a37d-debd...	3a56c204-4cad-43eb-b8a7-ddfde...	Yes	2025-08-19 14:52:38.002157+00
306eae89-d317-49ff-bf62-41b6ad8c4eb3	9f3a08d8-732f-4f05-85ed-cc118f...	ff051453-96b8-4b34-aaee-9adf27...	Maybe	2025-08-19 14:52:38.002157+00
37816032-30f1-49fa-8473-14716c2a85e8	bb3cb2cf-40e6-4cd2-b8e4-17bf2f...	112e79e9-e187-443a-9bec-c1a22a...	No	2025-08-19 14:52:38.002157+00
4c1aa0d4-41cf-41a8-ad3e-79aaeb04c0f4	9f3a08d8-732f-4f05-85ed-cc118f...	112e79e9-e187-443a-9bec-c1a22a...	Yes	2025-08-19 14:52:38.002157+00

Users:



id uuid	name text	email text	created_at timestamptz
014c0b82-b0b1-439c-a37d-debd2a902af	Amit Rao	amit@example.com	2025-08-19 14:52:38.002157+00
100fd3b8-da46-402d-a7b0-3bc6a0c6bbc	Neha Gupta	neha@example.com	2025-08-19 14:52:38.002157+00
1a41daab-98b8-4a50-af7f-0b5a99b76490	Vikram Joshi	vikram@example.com	2025-08-19 14:52:38.002157+00
44559040-603a-457b-8c5c-7f76deee481	Priya Singh	priya@example.com	2025-08-19 14:52:38.002157+00

Design Choices:

- I used **UUIDs** as primary keys for `Users`, `Events`, and `RSVPs` because they provide globally unique identifiers.
- Each table includes a **`created_at` timestamp** to record when the record was inserted, which is a best practice in production systems.
- The `Events` table has a `created_by` field referencing the `Users` table, enforcing ownership of events.
- The `RSVPs` table uses foreign keys to link both `Users` and `Events`. I set **ON DELETE CASCADE** so that if a user or event is deleted, their RSVPs are also automatically removed, ensuring referential integrity.
- A **CHECK constraint** ensures that RSVP status can only be `Yes`, No, or `Maybe`.

Sample Data:

```sql
-- Users
INSERT INTO public.users (name, email) VALUES
('Alice Sharma', 'alice@example.com'),
('Rahul Verma', 'rahul@example.com'),
('Sneha Patel', 'sneha@example.com'),
('Arjun Mehta', 'arjun@example.com'),
('Priya Singh', 'priya@example.com'),
('Karan Kapoor', 'karan@example.com'),
('Riya Nair', 'riya@example.com'),
('Vikram Joshi', 'vikram@example.com'),
('Neha Gupta', 'neha@example.com'),
('Amit Rao', 'amit@example.com')
```

```
ON CONFLICT (email) DO NOTHING;

-- Events
INSERT INTO public.events (title, description, date, city, created_by) VALUES
('Tech Meetup',   'Networking and coding event', '2025-09-01', 'Bangalore', (SELECT
id FROM public.users WHERE email='alice@example.com')),
('Startup Pitch', 'Startup ideas presentation',  '2025-09-05', 'Hyderabad', (SELECT
id FROM public.users WHERE email='rahul@example.com')),
('AI Workshop',   'Hands-on AI training',        '2025-09-10', 'Mumbai',    (SELECT
id FROM public.users WHERE email='sneha@example.com')),
('Music Fest',    'Live music and fun',          '2025-09-15', 'Delhi',     (SELECT
id FROM public.users WHERE email='arjun@example.com')),
('Art Expo',      'Art and culture exhibition',  '2025-09-20', 'Chennai',   (SELECT
id FROM public.users WHERE email='priya@example.com'))
ON CONFLICT DO NOTHING;

-- RSVPs (20 deterministic rows, unique per user/event)
INSERT INTO public.rsvps (user_id, event_id, status) VALUES
((SELECT id FROM public.users WHERE email='alice@example.com'),  (SELECT id FROM
public.events WHERE title='Tech Meetup'),   'Yes'),
((SELECT id FROM public.users WHERE email='alice@example.com'),  (SELECT id FROM
public.events WHERE title='AI Workshop'),   'Maybe'),
((SELECT id FROM public.users WHERE email='rahul@example.com'),  (SELECT id FROM
public.events WHERE title='Tech Meetup'),   'No'),
((SELECT id FROM public.users WHERE email='rahul@example.com'),  (SELECT id FROM
public.events WHERE title='Startup Pitch'), 'Yes'),
((SELECT id FROM public.users WHERE email='sneha@example.com'),  (SELECT id FROM
public.events WHERE title='Tech Meetup'),   'Yes'),
((SELECT id FROM public.users WHERE email='sneha@example.com'),  (SELECT id FROM
public.events WHERE title='AI Workshop'),   'Yes'),
((SELECT id FROM public.users WHERE email='arjun@example.com'),  (SELECT id FROM
public.events WHERE title='Music Fest'),    'Maybe'),
((SELECT id FROM public.users WHERE email='arjun@example.com'),  (SELECT id FROM
public.events WHERE title='Startup Pitch'), 'No'),
((SELECT id FROM public.users WHERE email='priya@example.com'),  (SELECT id FROM
public.events WHERE title='Art Expo'),      'Yes'),
((SELECT id FROM public.users WHERE email='priya@example.com'),  (SELECT id FROM
public.events WHERE title='Music Fest'),    'Yes'),
((SELECT id FROM public.users WHERE email='karan@example.com'),  (SELECT id FROM
public.events WHERE title='Tech Meetup'),   'Maybe'),
((SELECT id FROM public.users WHERE email='karan@example.com'),  (SELECT id FROM
public.events WHERE title='Art Expo'),      'No'),
((SELECT id FROM public.users WHERE email='riya@example.com'),   (SELECT id FROM
public.events WHERE title='AI Workshop'),   'Yes'),
((SELECT id FROM public.users WHERE email='riya@example.com'),   (SELECT id FROM
public.events WHERE title='Startup Pitch'), 'Maybe'),
((SELECT id FROM public.users WHERE email='vikram@example.com'), (SELECT id FROM
public.events WHERE title='Startup Pitch'), 'Yes'),
((SELECT id FROM public.users WHERE email='vikram@example.com'), (SELECT id FROM
public.events WHERE title='Music Fest'),    'No'),
((SELECT id FROM public.users WHERE email='neha@example.com'),   (SELECT id FROM
public.events WHERE title='Art Expo'),      'Maybe'),
((SELECT id FROM public.users WHERE email='neha@example.com'),   (SELECT id FROM
public.events WHERE title='AI Workshop'),   'No'),
((SELECT id FROM public.users WHERE email='amit@example.com'),   (SELECT id FROM
public.events WHERE title='Tech Meetup'),   'Yes'),
((SELECT id FROM public.users WHERE email='amit@example.com'),   (SELECT id FROM
public.events WHERE title='Art Expo'),      'Yes
```

This assignment helped me learn how to design relational tables, apply constraints, enforce foreign keys, and use Supabase to manage a database in the cloud. The design ensures data consistency and scalability for real-world applications.