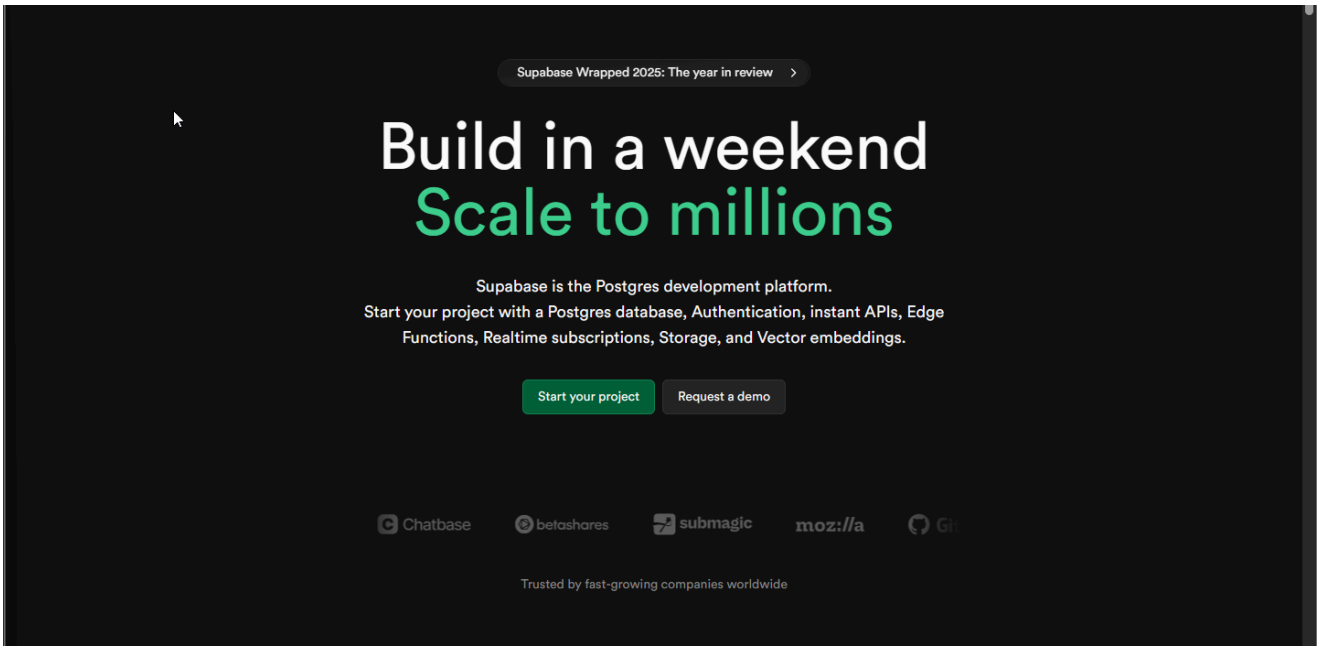
Supabase as Backend (Database Setup)

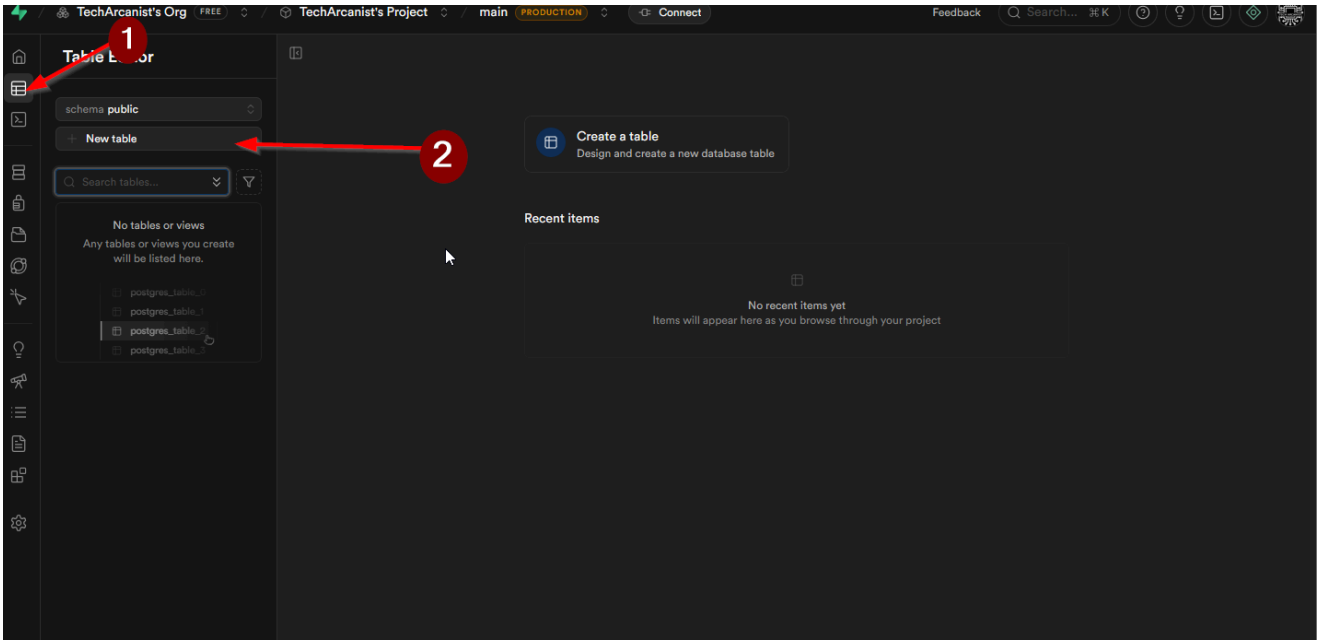
Now we will set up **Supabase** as the backend for our app.
We will create tables, define columns, and add relationships.

Step 1: Create Supabase Account and Project



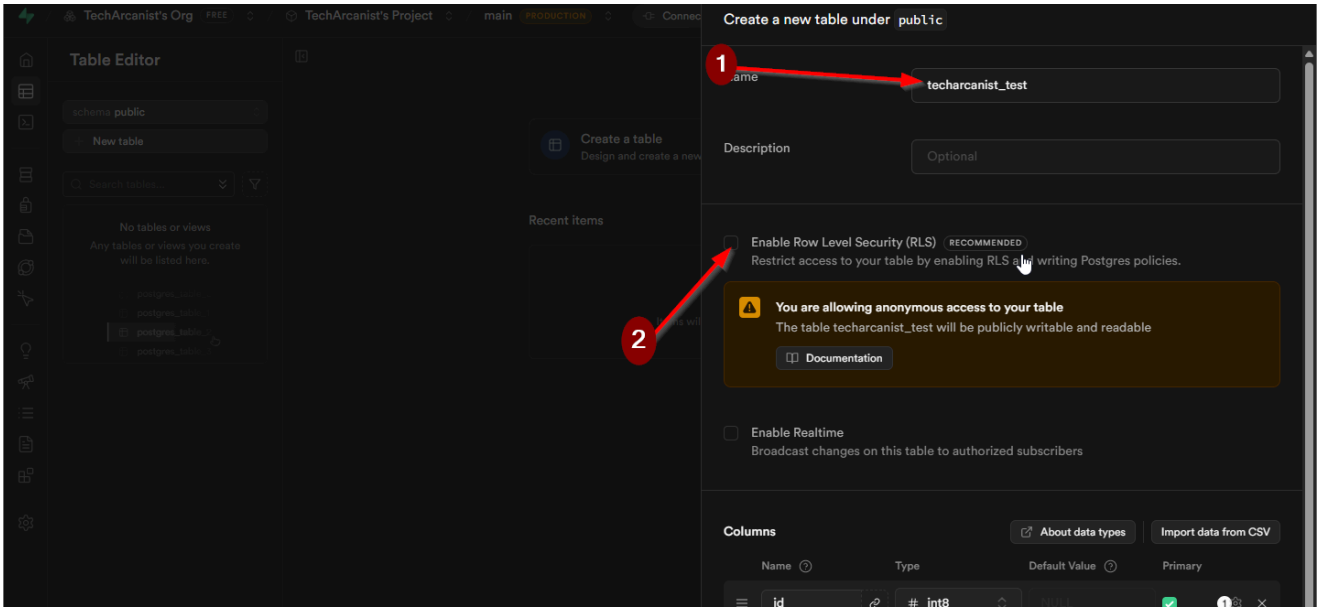
- Open **supabase.com**
- Create a new account
- Start a **New Project**

Step 2: Open Table Editor



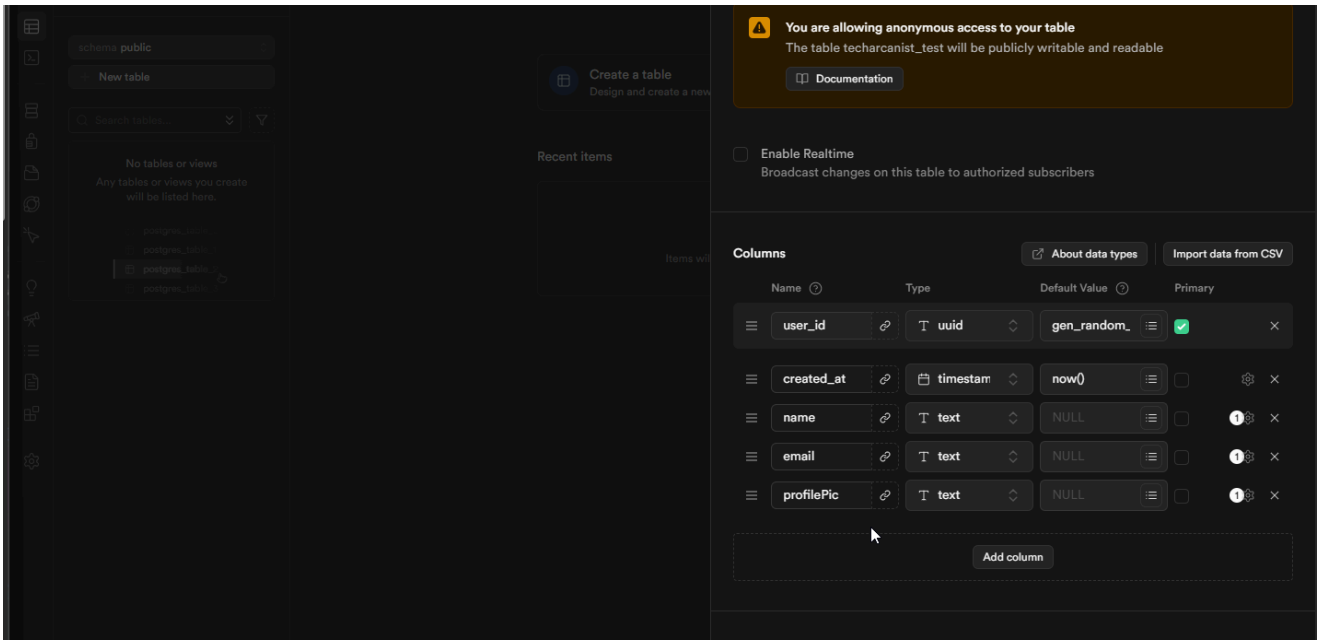
- Go to **Table Editor**
- Click on **New Table**

Step 3: Create User Table



- Name the table
- **Uncheck**: Enable Row Level Security (RLS)

Step 4: User Table Structure



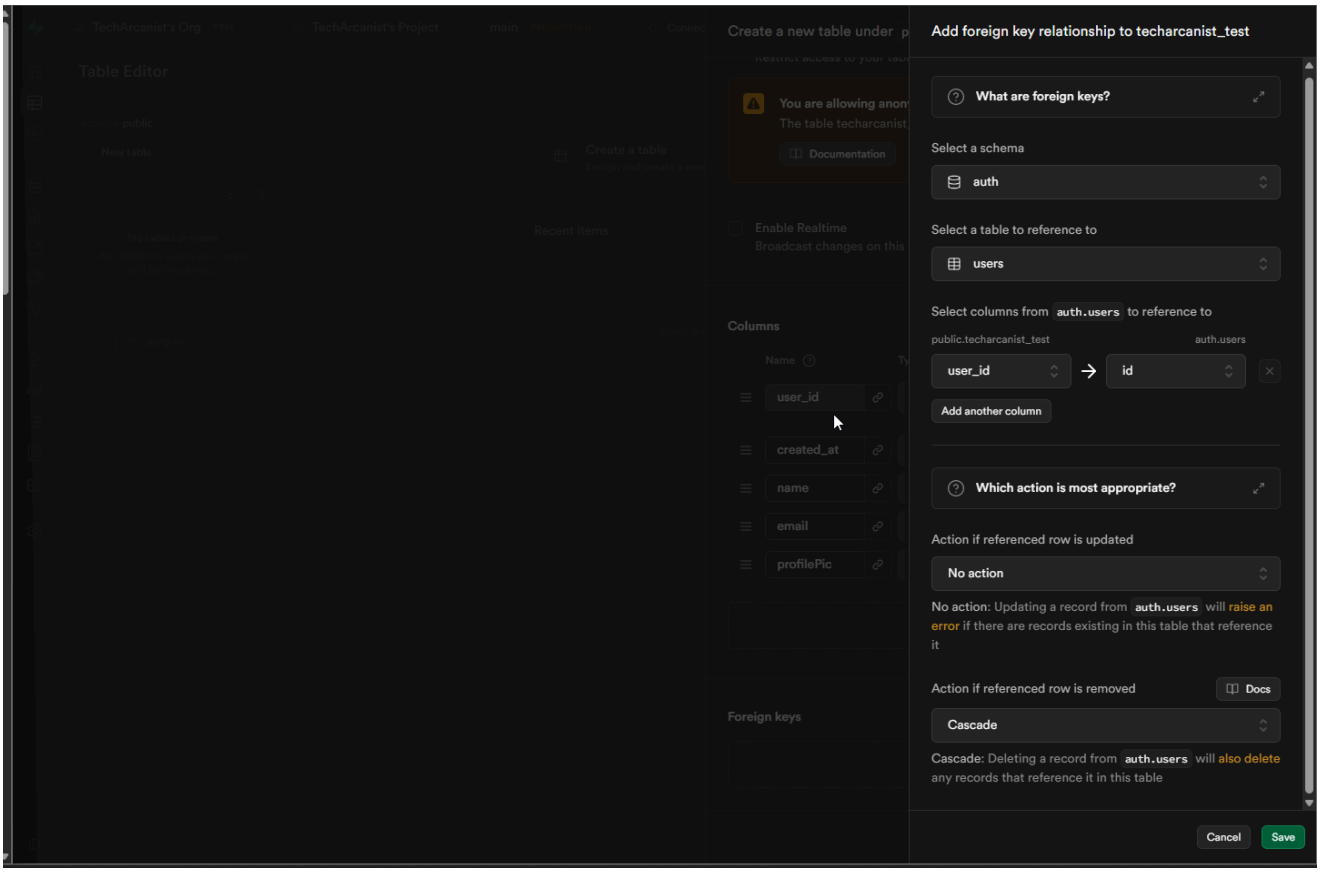
Create the table **techarcanist_test** with the following structure:

Table: `techarcanist_test`

Column Name	Data Type	Default Value	Primary Key
user_id	uuid	gen_random_uuid()	Yes
created_at	timestamp	now()	No
name	text	NULL	No
email	text	NULL	No
profilePic	text	NULL	No

This table will store user profile information.

Step 5: Add Foreign Key Relationship (User Auth)



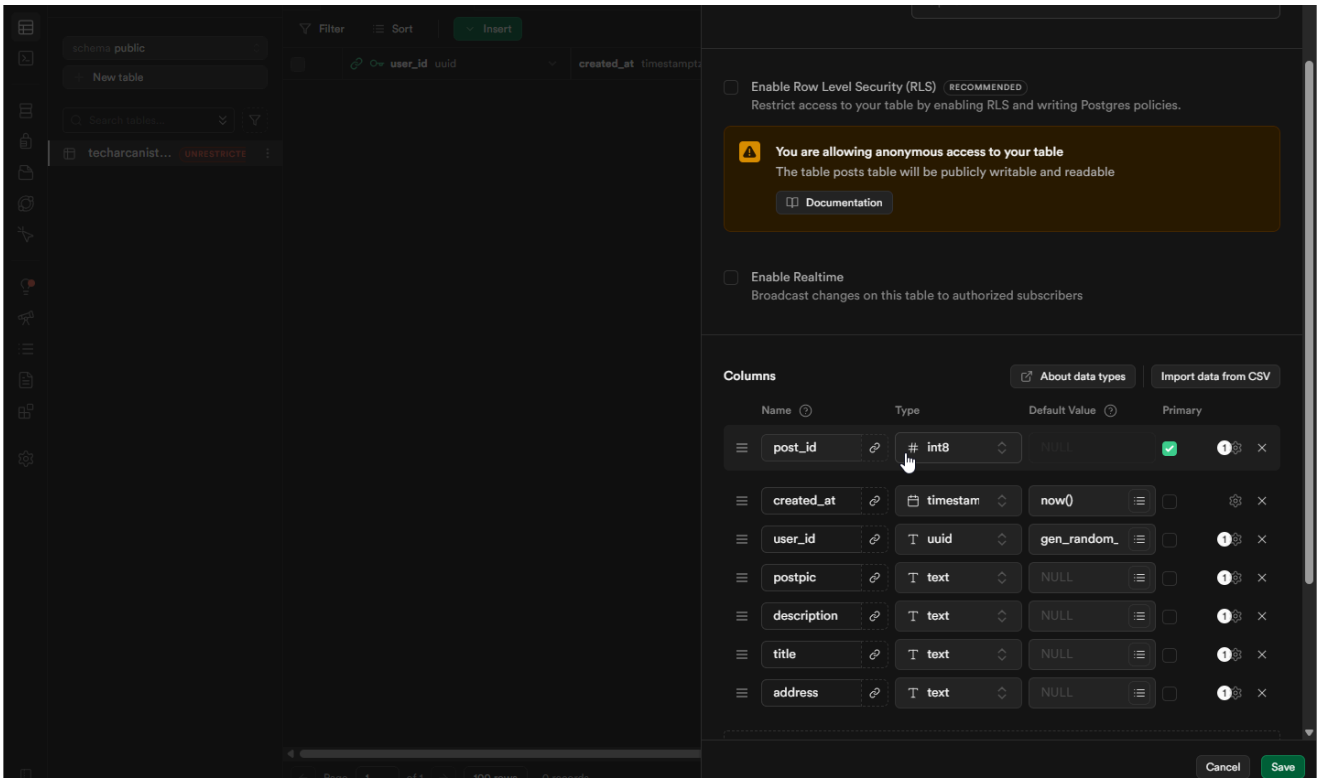
Add a foreign key so that **only real authenticated users** can exist in this table.

Set the relationship as:

- **Schema**: auth
- **Reference table**: users
- **Local column**: user_id
- **Reference column**: id
- **On update**: No action
- **On delete**: Cascade

This prevents fake users from using random user IDs.

Step 6: Create Posts Table



Create another table named **posts**
Again, **uncheck Row Level Security**.

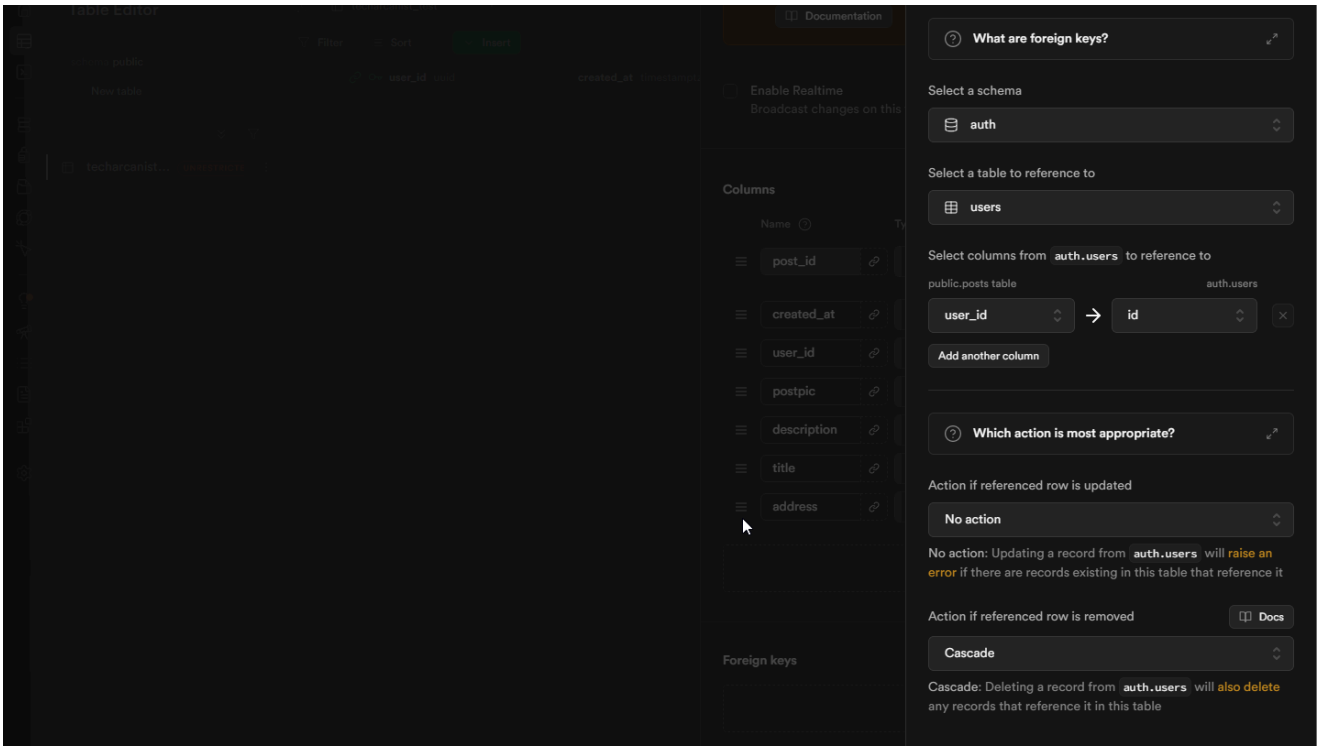
Table structure:

Table: `posts`

Column Name	Data Type	Default Value	Primary Key
post_id	int8	NULL	Yes
created_at	timestamp	now()	No
user_id	uuid	gen_random_uuid()	No
postpic	text	NULL	No
description	text	NULL	No
title	text	NULL	No
address	text	NULL	No

This table will store user posts.

Step 7: Add Foreign Key to Posts Table



Add a foreign key relationship for **user_id** in the posts table:

- Link it to the **auth.users.id**
- This ensures each post belongs to a valid authenticated user

---