

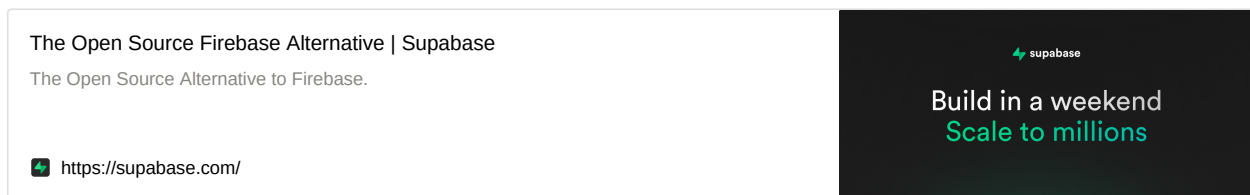
#3 Setting up Supabase Database

Introduction

Supabase is a popular open-source alternative to Firebase, which provides a suite of tools and services to build modern applications. One of the core features of Supabase is its database functionality, which allows developers to easily store and retrieve data in their applications. If you're looking to set up a Supabase database, this document will walk you through the steps required to achieve that.

Step 1: Sign up for a Supabase account

To get started with Supabase, you will need to sign up for a free account on their website. This process is very straightforward. Simply navigate to the Supabase homepage and click on the **"Start your project"** button. From there, follow the prompts to create an account and set up your profile.

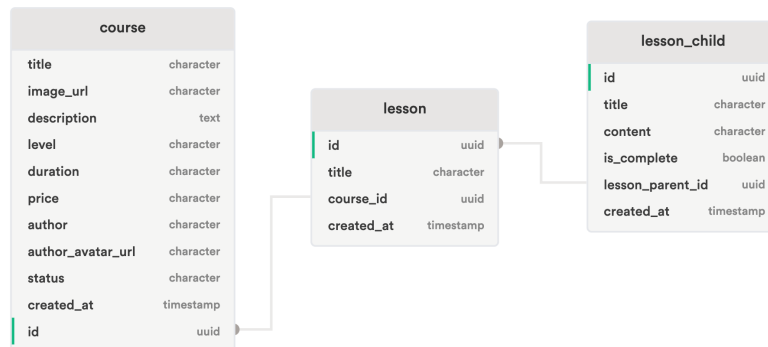


Step 2: Create a new project

Once you have set up your account, you will need to create a new project. From the Supabase dashboard, click on the **"New Project"** button and enter a name for your project. You will also need to select a region where your data will be stored. It is important to choose a region that is closest to your users, as this will help to reduce latency and improve the performance of your application.

<https://www.loom.com/embed/64d3a3365e02418d8e2368b2b1066d64>

Step 3: Create tables



Database Schema

To create the `course`, `lesson`, and `lesson_child` tables in your Supabase database, as shown in the database schema image above, you will need to use the following SQL definitions. By following these definitions, you can ensure that your tables will be created with appropriate attributes and constraints.

Copy and paste the SQL definitions into your Supabase SQL editor to create the necessary tables in your database.

Create a `course` table

```

create table
  public.course (
    id uuid not null default uuid_generate_v4 (),
    title character varying null,
    image_url character varying null,
    description text null,
    level character varying null,
    duration character varying null,
    price character varying null,
    author character varying null,
    author_avatar_url character varying null,
    status character varying null,
    created_at timestamp with time zone null default now(),
    author_job character varying null,
    constraint course_pkey primary key (id)
  ) tablespace pg_default;

```

The `public.course` table will store information about each course in your Supabase database. It contains fields such as the course title, image URL, description, level, duration, price, author, author avatar URL, status, creation date, and author job. A primary key constraint is also applied on the `id` field to ensure each course has a unique identifier.

<https://www.loom.com/embed/784cadecca45402cba2b2cd298bb386c>

Create a `lesson` table

```
create table
public.lesson (
  id uuid not null default uuid_generate_v4 (),
  title character varying null,
  course_id uuid null,
  created_at timestamp with time zone null default now(),
  constraint lesson_pkey primary key (id),
  constraint lesson_course_id_fkey foreign key (course_id) references course (id) on delete cascade
) tablespace pg_default;
```

The `public.lesson` table is used to store information about each lesson in your Supabase database. It includes fields such as the lesson title, course ID, and creation date. A primary key constraint is added to the `id` field, and a foreign key constraint is applied on the `course_id` field to ensure each lesson is associated with the correct course in the `public.course` table.

<https://www.loom.com/share/f00e161fcb3d47b8829ed86b3a4e7621>

Create a `lesson_child` table

```
create table
public.lesson_child (
  id uuid not null default uuid_generate_v4 (),
  title character varying null,
  content character varying null,
  is_completed boolean null default false,
  lesson_parent_id uuid null,
  created_at timestamp with time zone null default now(),
  constraint lesson_child_pkey primary key (id),
  constraint lesson_child_lesson_parent_id_fkey foreign key (lesson_parent_id) references lesson (id) on delete cascade
) tablespace pg_default;
```

Finally, the `public.lesson_child` table is used to store information about each child lesson in your Supabase database. It includes fields such as the lesson child title, content, completion status, and parent lesson ID. A primary key constraint is applied on the `id` field, and a foreign key constraint is added on the `lesson_parent_id` field to ensure each child lesson is associated with the correct parent lesson in the `public.lesson` table.

<https://www.loom.com/share/9214260f34b24d409bae062095840428>

Conclusion

If we have followed the steps outlined in this document, we should now have a fully functional Supabase database that is ready to be integrated into our application.