NOVI Hogeschool — AI Leerlijn

Les 8

Van In-Memory naar Supabase

Database koppelen aan Next.js + /create pagina bouwen

Les	8 van 18
Duur	3 uur (09:00 - 12:00)
Onderwerp	Supabase koppelen + /create pagina
Vereisten	Werkend QuickPoll + Supabase project

Inhoudsopgave

Deel 1: Supabase koppelen aan Next.js

Live Coding — docent toont, studenten volgen mee

Stap 1.1 — Supabase client library installeren

Installeer de Supabase JavaScript client library in je project:

```
npm install @supabase/supabase-js
```

Stap 1.2 — Environment variables instellen

Voeg je Supabase credentials toe in **.env.local**:

```
NEXT_PUBLIC_SUPABASE_URL=https://your-project.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=your-anon-key-here
```

> *Vervang de URL en key met je eigen Supabase project credentials. Na .env wijzigen moet je de dev server herstarten!*

Stap 1.3 — Supabase client aanmaken

Maak **lib/supabase.ts** aan:

```
import { createClient } from "@supabase/supabase-js";

export const supabase = createClient(
    process.env.NEXT_PUBLIC_SUPABASE_URL!,
    process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!
);
```

Stap 1.4 — Types updaten

Update **types/index.ts** met Poll en Option interfaces:

```
export interface Poll {
    id: number;
    question: string;
    created_at: string;
    options: Option[];
}

export interface Option {
    id: number;
    poll_id: number;
    text: string;
    votes: number;
    created_at: string;
}
```

Stap 1.5 — Supabase queries in data.ts

Herschrijf **lib/data.ts** met Supabase queries:

```
import { supabase } from "./supabase";
import { Poll } from "@/types";

export async function getPolls(): Promise<Poll[]> {
    const { data, error } = await supabase
        .from("polls")
        .select("*, options(*)");

    if (error) {
        console.error("Error fetching polls:", error);
        return [];
    }

    return data || [];
}
```

```
export async function getPollById(id: number): Promise<Poll | null> {
    const { data, error } = await supabase
        .from("polls")
        .select("*, options(*)")
        .eq("id", id)
        .single();

    if (error) return null;
    return data;
}
```

```
export async function votePoll(optionId: number): Promise<boolean> {
    const { error } = await supabase.rpc("vote_option", {
        option_id: optionId
    });

    if (error) {
        console.error("Error voting:", error);
        return false;
    }

    return true;
}
```

Stap 1.6 — Homepage server component

Update **app/page.tsx** als async server component:

```
import { getPolls } from "@/lib/data";
import Link from "next/link";
import PollItem from "@/components/PollItem";

export default async function HomePage() {
    const polls = await getPolls();

    return (
        <div className="w-full max-w-2xl mx-auto p-6">
            <h1 className="text-3xl font-bold mb-6">Huidige Polls</h1>
            <Link href="/create" className="text-blue-600 hover:underline mb-6 block">
                + Nieuwe Poll
            </Link>
            {polls.map((poll) => (
                <PollItem key={poll.id} poll={poll} />
            ))}
        </div>
    );
}
```

Stap 1.7 — PollItem Component

Maak **components/PollItem.tsx** ('use client') met percentage bars per optie. Elke optie linkt naar /poll/[id] voor stemmen.

Stap 1.8 — VoteForm Component

Maak **components/VoteForm.tsx** ('use client') dat opties toont als knoppen. Bij klik: roep votePoll() aan en toon feedback.

Stap 1.9 — Poll Detail Pagina

Maak **app/poll/[id]/page.tsx** (async server component) dat de poll ophaalt via getPollById() en de VoteForm component weergeeft.

Stap 1.10 — API Route

Maak **app/api/polls/[id]/route.ts** met GET en POST handlers. POST verwerkt stemmingen en retourneert de bijgewerkte poll.

Deel 2: /create pagina bouwen

Zelf Doen — 60 minuten

INSERT queries

Om een nieuwe poll aan te maken gebruik je twee INSERT queries:

```
// 1. Insert poll
const { data: poll } = await supabase
    .from("polls")
    .insert({ question: "Wat is je favoriete taal?" })
    .select()
    .single();

// poll is nu { id: 42, question: "Wat is je favoriete taal?", ... }

// 2. Insert options (meerdere tegelijk)
await supabase.from("options").insert([
    { poll_id: poll.id, text: "JavaScript", votes: 0 },
    { poll_id: poll.id, text: "Python", votes: 0 },
    { poll_id: poll.id, text: "Rust", votes: 0 }
]);
```

> *.insert() = INSERT statement. .select().single() = geef terug wat je net inserted. poll.id gebruiken we voor de options.*

RLS INSERT policy

Voeg deze policies toe in Supabase SQL Editor zodat inserts werken:

```
-- INSERT policy voor polls
CREATE POLICY "Allow public insert on polls"
ON polls FOR INSERT
TO anon
WITH CHECK (true);

-- INSERT policy voor options
CREATE POLICY "Allow public insert on options"
ON options FOR INSERT
TO anon
WITH CHECK (true);
```

> *Dit is tijdelijk — volgende les beperken we dit met Auth (alleen ingelogde users).*

Opdracht: Bouw de /create pagina

Maak **app/create/page.tsx** met de volgende vereisten:

1. 'use client' bovenaan (we hebben useState en event handlers nodig)
2. Text input voor de vraag
3. Meerdere text inputs voor opties (minimaal 2)
4. "+ Optie toevoegen" knop om meer opties toe te voegen
5. "Poll aanmaken" submit knop
6. Bij submit: INSERT in polls, dan INSERT in options, dan redirect naar /
7. Error handling: toon foutmeldingen als INSERT mislukt

Hints

• Gebruik **useState** voor question (string) en options (string[])
• Gebruik **useRouter** van next/navigation voor redirect
• Vergeet **e.preventDefault()** niet in handleSubmit
• Na insert: **router.push("/")** om terug te gaan
• Test in Supabase dashboard of je poll echt in de database staat

Huiswerk

Verplicht:

1. **/create pagina afmaken** als je niet klaar was in de les
2. **Validatie toevoegen:**
   - Vraag mag niet leeg zijn
   - Opties moeten uniek zijn
   - Minimaal 2 opties
   - Error messages tonen aan gebruiker

Extra (voor snelle studenten):

1. **Delete functionaliteit** — delete knop bij elke poll, verwijder poll + opties
2. **SQL queries** — test queries direct in Supabase SQL Editor
3. **Realtime subscriptions** — bekijk Supabase realtime docs
4. **Styling verbeteren** — maak de app mooier met Tailwind

Volgende les: Supabase Auth — inloggen, registreren, en bepalen wie wat mag doen!