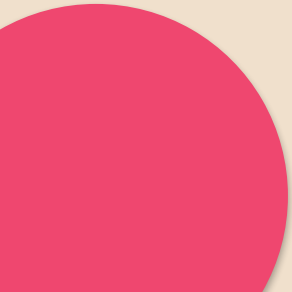




Les 7

Supabase Setup

Van in-memory naar een echte Postgres database



Les 5+6 - QuickPoll

Wat draait

- v Next.js QuickPoll app
- v Server + Client Components
- v Routing + dynamic /poll/[id]
- v Server Actions voor voting
- v Loading + error + not-found

Wat ontbreekt

- x Data verdwijnt bij restart
- x Stemmen worden niet bewaard
- x Geen echte database
- x Niet productie-ready
- x Geen multi-user logica

Vandaag: echte database.

Tool: Supabase - Postgres + Auth + Storage + Realtime in een dashboard.

Open-source, gratis tier, zeer geschikt voor leren + kleine projecten.

Eind van les: homepage leest polls vanuit echte Postgres-DB.

Planning

Vandaag - 180 minuten

Welkom + terugblik 10 min

Wat is Supabase + Postgres basics 20 min

Account + project aanmaken 15 min

LIVE DEMO 1 - Tabellen + relaties 20 min

Pauze 15 min

LIVE DEMO 2 - SQL editor + RLS intro 20 min

LIVE DEMO 3 - Client + eerste SELECT 30 min

Zelf koppelen aan jouw QuickPoll 40 min

Huiswerk + afsluiting 10 min

Theorie 1

Wat is Supabase?

Open-source Backend-as-a-Service. Een dashboard, veel features:

- | | | |
|---|--------------------------|---|
| 1 | Postgres database | Echte SQL, geen NoSQL |
| 2 | Auth | Login, signup, magic link, OAuth (Les 18) |
| 3 | Storage | Files uploaden |
| 4 | Realtime | Subscribe op DB changes |
| 5 | Edge Functions | Serverless deno-runtime |
| 6 | Dashboard | Tabel-editor, SQL editor, auth-management |

Open-source, self-hostbaar als je wilt - geen vendor lock-in.

Voor leerlijn: gratis tier - 500 MB DB, 1 GB storage, 50k auth users.

Alternatives: Firebase (NoSQL), Planetscale (MySQL), Neon (Postgres).

Setup

Account + project aanmaken

- 1 supabase.com -> Sign up (mag met GitHub)
- 2 New project
- 3 Naam: quickpoll
- 4 Database password (sla op!)
- 5 Region: Europe (Frankfurt)
- 6 Plan: Free
- 7 Wacht 1-2 min op provisioning

Settings -> API geeft je:

Project URL	<code>https://abc123.supabase.co</code>
anon key	publiek - mag in client
service_role key	ADMIN - NIET in client!

Theorie 2

Postgres basics

Relationele SQL database. Industrie-standaard sinds 1986.

polls

+-----+-----+-----+-----+		+-----+-----+-----+-----+	
id	title		created_at
+-----+-----+-----+-----+			
1	Beste taal?		2026-06-04
2	Beste IDE?		2026-06-04

Veelgebruikte types:

bigserial

Auto-increment ID

uuid

Random UUID (gen_random_uuid())

text

Variabele tekst

integer

Geheel getal

boolean

true/false

timestamp

Datum + tijd

jsonb

JSON (queryable)

Constraints: primary key, not null, unique, references (FK), default.

LIVE DEMO 1

Tabellen + relaties - ~20 min

Twoe opties: Table Editor (UI) of SQL Editor (productie):

```
create table polls (  
  id          bigserial primary key,  
  title       text not null,  
  created_at  timestamp default now()  
);
```

```
create table options (  
  id          bigserial primary key,  
  poll_id     bigint not null references polls(id) on delete cascade,  
  text        text not null,  
  votes       integer default 0  
);
```

on delete cascade: poll weg -> opties automatisch weg. One-to-many.

LIVE DEMO 2

SQL editor + RLS intro - ~20 min

SQL editor - eerste queries:

```
insert into polls (title) values ('Beste programmeertaal?');  
  
select * from polls order by created_at desc limit 10;  
update polls set title = 'Nieuwere titel' where id = 1;  
delete from polls where id = 1;
```

RLS - Row Level Security (intro):

```
alter table polls enable row level security;  
create policy "demo open" on polls for all to anon  
  using (true) with check (true);
```

Zonder RLS = iedereen alles. Voor nu: open policies. Diep in Les 18.



Pauze

15 minuten

Setup

Supabase client in Next.js

```
pnpm add @supabase/supabase-js
```

`.env.local` (in root):

```
NEXT_PUBLIC_SUPABASE_URL=https://abc123.supabase.co  
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJ...
```

`lib/supabase.ts`:

```
import { createClient } from "@supabase/supabase-js";  
  
export const supabase = createClient(  
  process.env.NEXT_PUBLIC_SUPABASE_URL!,  
  process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!,  
);
```

NEXT_PUBLIC_ prefix want anon key mag in client (RLS beschermt data).

LIVE DEMO 3

Eerste SELECT vanuit Server Component - ~30 min

```
// app/page.tsx
import { supabase } from "@/lib/supabase";

export default async function Home() {
  const { data: polls, error } = await supabase
    .from("polls")
    .select("*")
    .order("created_at", { ascending: false });

  if (error) return <p>Error: {error.message}</p>;

  return (
    <ul>
      {polls?.map((p) => <li key={p.id}>{p.title}</li>)}
    </ul>
  );
}
```

Wat hier gebeurt:

Server fetcht data -> HTML naar client. Geen API route, geen useEffect.

Bonus

Polls + options samen ophalen

Supabase JS doet automatisch joins via select():

```
const { data } = await supabase
  .from("polls")
  .select("*", options(*));
```

Resultaat (typed):

```
[
  {
    id: 1,
    title: "Beste taal?",
    options: [
      { id: 1, text: "JS", votes: 10 },
      { id: 2, text: "Python", votes: 5 }
    ]
  }
]
```

SQL onder de motorkap. JS API maakt 't simpel.

LESOPDRACHT

Koppel Supabase aan jouw QuickPoll (~40 min)

Checklist:

- v Supabase project aangemaakt + region NL
- v Tabellen polls + options met FK + cascade
- v RLS aan, open policy voor anon
- v Een paar test-polls + opties via SQL insert
- v lib/supabase.ts en .env.local goed
- v app/page.tsx leest polls vanuit DB
- v Optioneel: poll detail page leest opties

Tips:

- Cursor + AGENTS.md: voeg Supabase patterns toe
- Check de docs - quickstart voor Next.js
- Vergeet .env.local niet in .gitignore (al default)

Huiswerk

Voor Les 8 (CRUD + mutations)

1. Supabase aan QuickPoll gekoppeld

Lijst polls vanuit DB op homepage

2. SQL leren - 3 oefeningen

SELECT + WHERE + JOIN op je eigen data

3. Lees Supabase Next.js quickstart

supabase.com/docs - Next.js sectie

4. Inleveren via Teams

GitHub repo + screenshot van werkende DB-fetch

Tip: sqlbolt.com - korte interactieve SQL crash course.

Volgende les

Les 8 - CRUD + Mutations

Van read-only naar volledige app:

- INSERT - poll-creation form werkend
- UPDATE + DELETE
- Server actions met Supabase
- Voting via RPC (stored procedure)
- TanStack Query voor client-side data
- Types genereren uit DB schema

Vorbereiden:

Lesopdracht af - QuickPoll leest minstens een poll vanuit Supabase.

Vragen?

Tot volgende week!