

# Innhold

<b>1</b>	<b>Databaseoversikt FastNotes</b>	<b>1</b>
<b>2</b>	<b>Sekvensdiagram for Opprette nytt notat</b>	<b>3</b>

# Figurliste

1	ER-diagram for databasen (Supabase). . . . .	2
2	Sekvensdiagram som viser interaksjon mellom app og database ved opprettelse av notat. . . . .	3

# 1 Databaseoversikt FastNotes

## Oversikt

Prosjektet bruker Supabase som backend. Autentisering håndteres av Supabase sin innebygde tabell `auth.users`. Notater lagres i tabellen `notes`.

## Tabeller

### `auth.users` (Supabase Auth)

Inneholder brukere som kan logge inn. Viktigste felt for prosjektet er:

- `id` (UUID, primærnøkkel) – unik bruker-ID
- Supabase lagrer også f.eks. e-post og tidspunkt

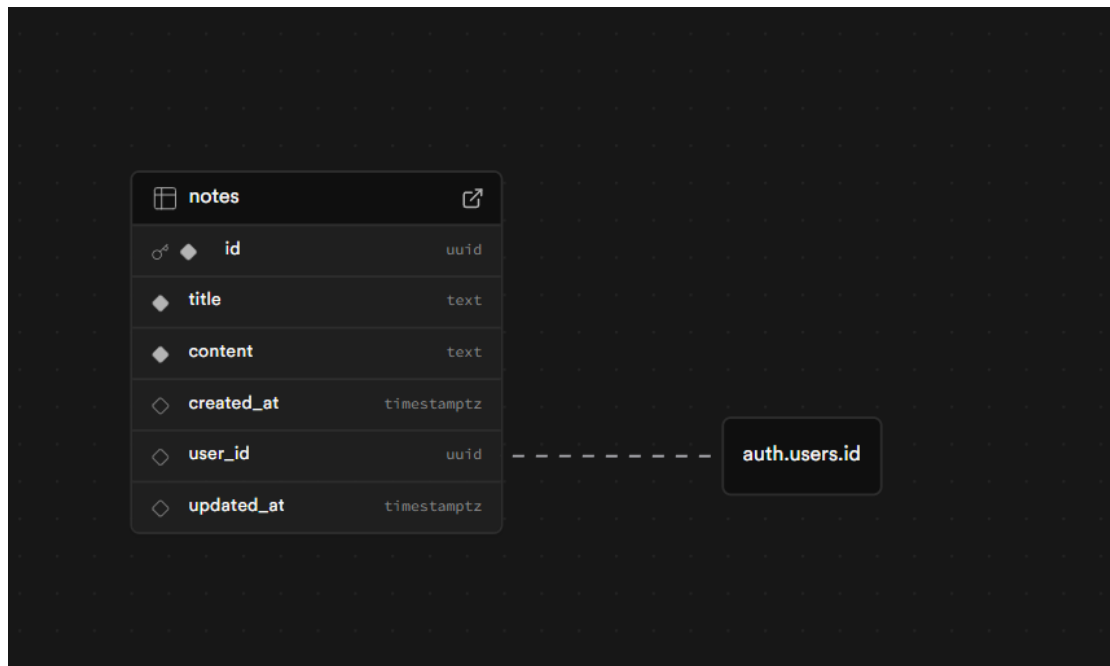
### `notes`

Lagrer notater som tilhører en bruker:

- `id` (UUID, PK) – unik ID for notatet
- `title` (text) – tittel
- `content` (text) – innhold
- `created_at` (timestampz) – opprettet tidspunkt
- `updated_at` (timestampz) – sist oppdatert
- `user_id` (UUID, FK) – peker til `auth.users.id`

## Relasjon

Relasjonen er **1:N**: en bruker kan ha mange notater, og hvert notat tilhører kun en bruker via `notes.user_id`.

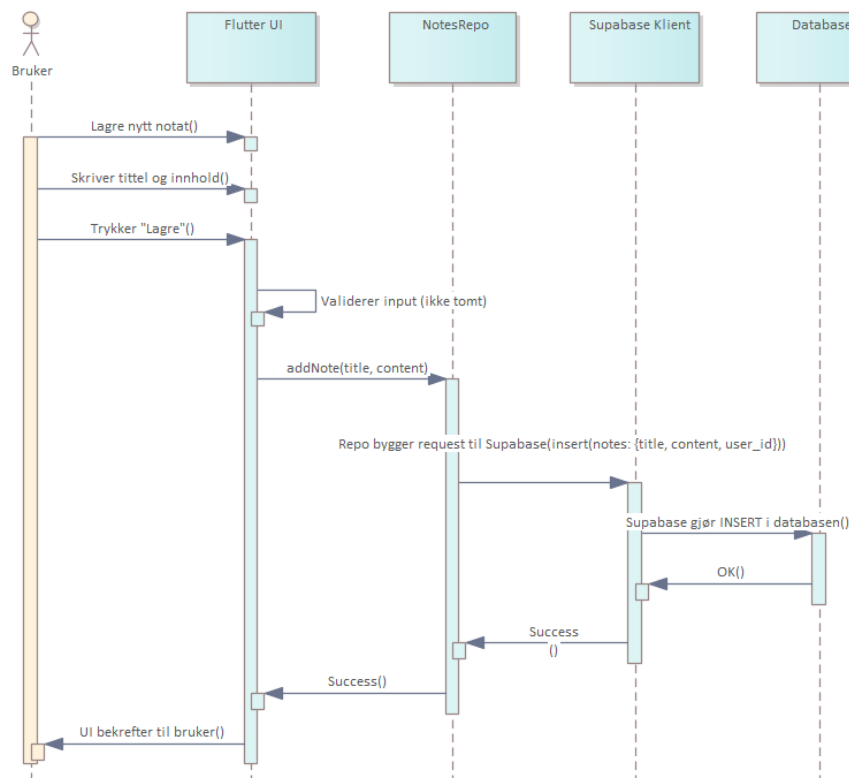


Figur 1: ER-diagram for databasen (Supabase).

## 2 Sekvensdiagram for Opprette nytt notat

Sekvensdiagrammet viser hvordan appen kommuniserer med databasen når en bruker oppretter et nytt notat.

Prosessen starter når brukeren trykker på Lagre. Flutter UI validerer input og kaller **NotesRepo**, som sender en insert forespørsel via Supabaseklienten til databasen. Databasen lagrer notatet og returnerer en bekreftelse. Svaret sendes tilbake gjennom lagene, og brukeren får en visuell bekreftelse på at notatet er lagret.



Figur 2: Sekvensdiagram som viser interaksjon mellom app og database ved opprettelse av notat.