**Engineer** Data

{ Databasedesign og datamodellering }

# Et fornuftig sted for dataene dine

En struktur som samsvarer med hvordan virksomheten faktisk fungerer, slik at spørsmålene du trenger svar på senere, fortsatt er mulige å stille.

## Avklarer dere fortsatt hva dere trenger?

Fortell oss hva som går galt, eller hva dere prøver å bygge. Vi sier hva det vil kreve, omtrent hva det vil koste, og om vi er de rette til oppgaven.

**Fra 1 400 DKK timen, eksklusive mva. — og mindre per time ved et lengre oppdrag.**

**Ta kontakt** Se hva det koster

Utvalgt arbeid som viser denne tjenesten.

### Gjør tjenesten til et resultat

La «Databasedesign og datamodellering» støtte et forretningsmål. Vi kan knytte det tekniske arbeidet til brukerne, driften og beslutningene det skal støtte. En struktur som samsvarer med hvordan virksomheten faktisk fungerer, slik at spørsmålene du trenger svar på senere, fortsatt er mulige å stille.

**Ta kontakt**

### Undersøk det relevante arbeidet

Bruk arbeidet på denne siden til å undersøke tilnærmingen vår. Spør om avveiningene og forutsetningene bak hvert resultat. Designet og driftet 6 ETL/ELT-pipelines — 6 ETL-pipelines på tvers av tre databasemotorer

**Designet og driftet 6 ETL/ELT-pipelines** Ta kontakt →

### Designet og driftet 6 ETL/ELT-pipelines

Designet, implementerte og administrerte 6 ETL/ELT-pipelines med Google BigQuery, MSSQL, PostgreSQL, Shell-skripting, PL/pgSQL og Transact-SQL, og integrerte data for effektiv behandling i et Python-API.

### Drev 100 datavarehus: Postgres, MS SQL, BigQuery

Utformet, opprettet og administrerte 100 databaser for datavarehus i PostgreSQL, MS SQL og Google BigQuery, hovedsakelig med GIS- og tidsseriedata, og optimaliserte ytelse og skalerbarhet.

### En maritim plattform med klare systemgrenser

Utformet arkitekturen for en lagdelt maritim plattform som skiller en Next.js PWA-frontend, et Go-API (Huma/Fiber) og et funksjonslag i PostgreSQL, med all forretningslogikk i databasen.

### Designet JSON rett fra Postgres til API-et

Designet en JSON-passthrough-arkitektur der PostgreSQL-funksjoner returnerer komplett JSON som Go-API-et videresender uendret, noe som fjernet mellomliggende unmarshalling og frikoblet frontend fra skjemaendringer.

### Designet et funksjonsbasert datalag i PostgreSQL

Designet et funksjonsbasert datalag i PostgreSQL — 1 275 lagrede funksjoner fordelt på 34 skjemaer — slik at hver lesing og skriving går gjennom en funksjon databasen kan gi tilgang til, i stedet for gjennom en tabell.

### Tidsordnede nøkler for en voksende datamodell

Tok i bruk tidsordnede UUID v7-identifikatorer (PostgreSQL 18) som entitetsnøkler for å redusere fragmenteringen i B-tre-indekser og gjøre spørringene raskere.

### Implementerte en katalogstyrt deep merge

Implementerte en katalogstyrt deep merge av lagrede JSON-innstillinger, som forhindrer krasj ved manglende nøkler når skjemaet utvikler seg.

### Modellerte 348 tabeller i 34 Postgres-skjemaer

Modellerte det maritime domenet i 348 normaliserte tabeller fordelt på 34 PostgreSQL-skjemaer — fagfolk, selskaper, skip, stillinger, omtaler og resten — med SMALLINT-oppslag og UUID v7-nøkler.

### Bygde tilgangsrettigheter for Stripe og appkjøp

Bygde laget for betaling og tilgangsrettigheter — Stripe sammen med kjøp i appen hos Apple og Google — som styrer tilgangen til katalogen, søket og eksporten gjennom en tilgangsmodell på 11 tabeller som kontrolleres på serveren.

### Flyttet tregt arbeid over i en River-jobbkø

Flyttet tregt arbeid ut av forespørselsbanen og over i en River-jobbkø — 15 worker-moduler, 8 planlagte oppgaver og 20 pg_cron-jobber — slik at en forespørsel returnerer mens arbeidet bak den fortsetter.

### Bygde egen analyse inne i PostgreSQL

Bygde egen produktanalyse i PostgreSQL — 47 funksjoner over partisjonerte hendelsestabeller som rydder seg selv — pseudonymisert bak et roterende salt og betinget av den besøkendes samtykke.

### Bygde en lojalitetslommebok, ligaer og merker

Bygde lojalitets- og omdømmesystemet — 67 funksjoner over en hovedbok på 31 tabeller, med ligaer, merker og en innløsningsbutikk — med radlås på saldoen for å lukke vinduet for dobbeltforbruk.

### Bygde arbeidsflater for rekruttering og sjøfolk

Bygde arbeidsflater for rekruttering og sjøfolks karriere; det registrerte øyeblikksbildet av implementasjonen omfatter 151 lagrede funksjoner i 48 tabeller for stillinger, søknader, sertifikater, rangutvikling og fartstid.

### Bygde en databasedrevet CV-generator

Bygde en databasedrevet generator for CV, referanser, portefølje og søknadsbrev i Python — 41 moduler, 10 580 linjer — som gjengir seks utdataformater fra én SQLite-kilde med 23 tabeller, satt sammen av en idempotent pipeline i 19 trinn.

## Leter dere etter Databasedesign og datamodellering?

Bruk de tilknyttede arbeidene til å vurdere relevant erfaring. Fortell oss hva dere skal bygge; vi kan diskutere omfang, fremgangsmåte og overlevering.

**Ta kontakt** Hele porteføljen

### Avgrens oppgaven før bestilling

Ta med et representativt datasett, spørsmålene det skal besvare og dagens flaskehals. Vi kan avtale målinger av kvalitet, gjennomstrømning og gjenoppretting.

**Se hva det koster** Ta kontakt →

# Gå gjennom kravene med oss

Fortell hva dere trenger innen «Databasedesign og datamodellering», inkludert fristen og teamet som skal vedlikeholde løsningen. Vi kan drøfte en praktisk vei videre.

**Ta kontakt**

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

**Laget med kjærlighet av D.**

**CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N**

**Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.**

**Skrevet ut fra https://data.engineer.company/nb/services/database-design/**