

## Data Usability

### Etablering av business intelligence-løsninger med høy brukskvalitet

Usability er et sentralt tema innen applikasjonsutvikling og dreier seg om hvor brukervennlig, intuitiv og effektiv applikasjonen er i forhold til intensjonen. Brukskvalitet er en ofte brukt oversettelse. Innen business intelligence kommer dette temaet i hovedsak opp i forbindelse med rapporteringsfunksjonalitet og visualisering. Men hovedingrediensen i alle BI-løsninger er selve informasjonen – dataene. I en vellykket BI-løsning brukes de samme dataene på tvers av forretningsprosesser og organisatoriske skillelinjer i virksomheten. I rapporter og analyser er selve dataene den synligste komponenten, og manglende datakvalitet vil resultere i lav brukskvalitet.

De fleste IT-prosjekter, også BI-prosjekter, defineres av funksjonelle krav. Dette kan være i form av definerte rapporter, skjermbilder eller annen type prosessstøtte. Databehovet som understøtter disse kravene er sjelden definert. Formålet med dette dokumentet er å belyse viktigheten av kravstilling til selve dataene og prosessen rundt dette med spesiell fokus på BI-løsninger. Hvordan skal vi tenke brukskvalitet i data-designet? Hvordan kan vi sikre gode datamodeller?

Et av de viktigste kjennetegnene ved en BI-løsning er stadige prosessendringer som påvirker bruksområder og grensesnitt. Selv om omfanget til en BI-implementering gjerne er avgrenset av et sett med definerte rapporter eller funksjoner, er det derfor ikke disse alene som avgjør brukskvaliteten.

Innen business intelligence dekkes mye av de funksjonelle kravene, for eksempel tabelloppstilling, grafer, mulighet for drill-down og slice&dice, gjennom standardverktøy. Men verktøyenes funksjonalitet vil i betydelig grad være avhengige av underliggende datadesign.

For å sikre data usability må vi fokusere på om datadesignet er:

- **Fleksibelt:** Er dataene godt tilrettelagt for analyse og ad-hoc rapportering?
- **Intuitivt:** Formidler dataenes navn og beskrivelse en effektiv og intuitiv forståelse av innholdet?
- **Konsistent:** Er dataene fullstendige og entydige på tvers av virksomheten?
- **Kvalitativt:** Er datakvaliteten kjent? Er kravene definert? Er dataene vasket/beriket?
- **Skalerbart:** Hvor enkelt det er å sammenstille dataene på andre måter til nye bruksområder?
- **Sporbart:** Hvor kommer dataene fra? Hvem produserer og vedlikeholder dem?
- **Tilgjengelig:** Er dataene tilgjengelige for brukerne, der de trenger den, når de trenger dem?

Det er helt essensielt å involvere brukerne i datadesignet. Det er brukernes begreper og definisjoner som skal benyttes. Vi skal i dette dokumentet se på prosessen for å sikre god brukerinteraksjon i modelleringsarbeidet. En grunnleggende teknikk er å skille mellom logiske og fysiske datamodeller. Den logiske modellen utvikles sammen med brukerne. I denne modellen holdes fokuset på brukernes databehov, og ingen tekniske betraktninger og hensyn tas på dette nivået. De tekniske elementene kommer i den fysiske modellen, som utvikles av dataarkitekten, gjerne i samarbeid med IT-tekniske brukere. På dette nivået legges nøkler og andre tekniske attributter inn i modellen. Andre tekniske betraktninger, som datamengder, endringstakt og ytelse, påvirker også den fysiske modellen.

I dette dokumentet skiller vi mellom flere sentrale roller:

- Dataarkitekten: Ansvarlig for design av datamodellene
- Brukere: Ikke-tekniske brukere av BI-løsningen
- Superbrukere: Analytikere og andre IT-tekniske brukere
- Andre interessenter: Har eierskap eller andre interesser til dataene, men har ingen definert rolle i prosjektet

Dataarkitekten er ansvarlig for dokumentasjon, datamodeller, fasilitere datamodelleringsprosessen og sikre forankring for beslutningene. Dataarkitekten må ha utdanning i datamodellering, men kan komme fra enten IT- eller brukersiden.

Vi skiller mellom brukere og superbrukere. Begge gruppene er brukere av BI-løsningen og tilhører forretningssiden, men superbrukerne har tyngre IT-teknisk forståelse. Vi kommer også til å se at det er viktig å identifisere og involvere andre datainteressenter i prosessen, blant annet fordi dataprodusentene i mange tilfeller sitter på utsiden av prosjektet.

## Datavarehus vs datamart

Virksomhetsomspennende BI-løsninger består som regel av flere datalag. Det er spesielt to av disse det er essensielt å forstå forskjellen på; datavarehus og datamart. Disse to lagene har i utgangspunktet forskjellige formål:

- Datavarehusets funksjon er å sikre at dataene er kvalitetssikret og generiske for virksomheten. Det betyr at dataene er konsistente og har en omforent form slik at de kan brukes på tvers av virksomheten (én felles sannhet).
- En datamart er tilegnet et spesifikt formål eller område og således utformet med dette som fokus. Virksomheten kan ha mange datamarter og datavarehuset er (i de aller fleste tilfeller) kilden til datamartene.

Det finnes forskjellige retninger innen utvikling av datalagene til BI-løsninger, såkalte referansearkitekturer. Vi tar ikke stilling til valg av referansearkitektur i dette dokumentet, men kommer til å ta utgangspunkt i de to nevnte lagene, som går igjen i de fleste referansearkitekturer. En modellform vi likevel kommer til å nevne i dette dokumentet, er dimensjonsmodellering, som i utgangspunktet tilhører retningen til Ralph Kimball. Dette fordi denne teknikken er vanlig å bruke på datamartlaget uansett valg av referansearkitektur. En signifikant forskjell er at Kimballs referansearkitektur kombinerer datavarehus og datamart i ett lag, en såkalt bussarkitektur, mens andre holder disse lagene adskilt.

Datavarehuset er et felles datalag for virksomhetens mange business intelligence behov. Dette laget skal ta vare på alle data som er underlag til de forskjellige datamartene. For å unngå å låse seg til rigide datamodeller, er det viktig å ligge unna funksjonelle krav på dette nivået og heller fokusere på hvilke data som er viktige å ta vare på og hvordan de logisk henger sammen. De funksjonelle behovene, som for eksempel spesifikasjon av en prosess eller rapport, blir kun et utgangspunkt for diskusjonen om hvilke data som trengs for å understøtte funksjonen – altså avgrensningen av omfanget til leveransen. Deretter må den funksjonelle spesifikasjonen legges til side, og fokuset flyttes til databehovet. I datavarehuslaget er det viktig å ta høyde for mulige endringer og fremtidige behov. Dette betyr for eksempel at selv om behovet i utgangspunktet er å beholde data på et aggregert nivå, vil det i dette laget likevel være riktig å ta inn dataene med finest mulig granularitet og med flere detaljer enn behovet i utgangspunktet tilsier. En datamart utledes som regel av et funksjonelt behov, som for eksempel støtte til en bestemt forretnings-

prosess. Dette kan for eksempel være kundesegmentering, kampanjestyring, prognostisering, etc. Disse spesifikke behovene vil i stor grad påvirke utformingen av datamodellen. Det er også viktig å designe datamarten på sluttbrukerapplikasjonens premisser. Så lenge datamarten holdes adskilt fra datavarehuset, er dette fullt akseptabelt (ved bruk av bussarkitektur må man i større grad tenke generisk ved design av datamarten). Så lenge det kun er ett miljø som skal benytte en datamart, kan vi også i større grad akseptere bruk av subkulturens begreper og definisjoner på dataene. Men datamarten bør i størst mulig grad bruke datavarehuset som kilde, så da er det viktig å dokumentere koblingen (mappingen) mellom datamartens særbegreper og datavarehusets generiske begreper.

Datamarter kan også benyttes til å begrense tilgangen til sensitive data, for eksempel ved at kun brukere ved HR-avdelingen får tilgang til den datamarten som inneholder detaljerte lønns- og personaldata. Det er ganske vanlig å styre alle brukertilganger på datamartnivå, mens få eller ingen brukere får tilgang direkte til datavarehuset.

## Identifiser brukere og interessenter

Det absolutt viktigste kriteriet for å sikre data usability er at dataenes navn og definisjoner har sitt opphav i brukernes eget begrepsapparat. Men det er ikke nok å lytte til én enkelt bruker eller ekspert. Et bredt utvalg av brukere og interessenter må involveres gjennom intervjuer og/eller workshops. For å identifisere hvilke personer som bør involveres i datadesignet, kan det være til god hjelp å etablere noen grove datakategorier (subjektområder) utledet fra en midlertidig forståelse av kravspesifikasjon og forretningsmodell. Subjektområdene kan for eksempel defineres ut fra de forretningsprosessene som er hovedkilden til dataene, og således være et godt utgangspunkt for hvor vi kan finne interessenter til dataene. Subjektområdene er avhengige av virksomhet og bransje, men noen eksempler kan være:

- Økonomi/regnskap
- Personal/lønn
- Kunder/kontakter/organisasjon
- Produkter/varianter/kategorier
- Salg/ordre/faktura
- Innkjøp/lager
- Produksjon

Merk at dette blir midlertidige kategorier, og kan endre seg etter hvert som forståelsen for dataene øker. Prosessen videre vil resultere i finere gruppeinndelinger innen for disse områdene. Disse grupperingene blir gjerne kalt subjekter eller logiske entiteter, avhengig av hvilken skole man tilhører. Vær oppmerksom på at dette ikke er det samme som fysiske entiteter (tabeller) slik de kommer til å bli implementert i databasen. Det fysiske designet vil som regel bli en IT-drevet aktivitet.

I datamartlaget er det også vanlig å dele inn entitetene i fakta (måltall/hendelser – det som skal telles eller summeres) og dimensjoner (grunndata – utvalgsområdene eller grupperingene til faktaene), i henhold til Ralph Kimballs lære om dimensjonsmodellering. Kimball introduserer også bruk av det han kaller en bussmatrise, som sammenkobler fakta og dimensjoner. Denne matrisen kan også være nyttig til å identifisere interessenter, da faktaene (hendelsene) ofte korresponderer med forretningsprosesser.

Fakta \ Dimensjon	Ansatt	Kanal	Konto	Kunde	Produkt
Aktivitet	X	X		X	
Balanse			X		X
Faktura				X	X
Innkjøp	X		X		X
Lønn					
Ordre	X	X		X	X
Regnskap			X		X

En viktig erkjennelse innen business intelligence er at brukerne av data som regel ikke er de samme som produserer eller vedlikeholder dataene. Dette betyr at vi må trekke inn personer som har eierskap og kjennskap til dataene, men som ikke nødvendigvis har interesser i BI-løsningen eller har dedikert tid til prosjektet. I slike tilfeller er det viktig med forankring høyt oppe i organisasjonen. Ofte vil BI-løsningen kunne gi muligheter for brukere ut over hovedinteressentene, og det kan være nyttig å synliggjøre mulighetene for å få engasjert dataprodusentene.

En CRUD-matrise (Create, Read, Update, Delete) kan være et nyttig instrument for å identifisere forretningsprosessene og hvilken rolle de har i forhold til de enkelte dataene. Eksempelet under viser en balansert målstyringsløsning, som har behov for et stort sett med data. Alle disse dataene produseres og vedlikeholdes i andre forretningsprosesser. CRUD-matrisen gir en god oversikt over hvilke miljøer vi må involvere i datadesignet.

Prosess Data	Balansert målstyring	CRM	Fakture- ring	HR	Kategori- styring	Marked	Ordre og kontrakt	Regnskap
Aktivitet	R	R				CRUD		
Ansatt	R			CRUD	R		R	R
Balanse	R				R			CRU
Faktura	R	R	CRUD					R
Innkjøp	R							
Kanal	R	R				CRUD		
Konto	R				R			CRU
Kunde	R	CRU	R			R	CRU	
Lønn	R			CRU				R
Ordre	R	CRUD	R				CRUD	
Produkt	R		R		CRUD			
Hovedbok	R							CR

## Ta utgangspunkt i funksjonelle behov

Datadesignet vil stort sett alltid ta utgangspunkt i et funksjonelt behov, for eksempel i form av en kravspesifikasjon. Databehovet vil i større eller mindre grad fremkomme av kravene, avhengig av spesifikasjonens detaljeringsgrad og hvorvidt en dataarkitekt eller analytiker har vært involvert i prosessen. Kravspesifisering av BI-løsninger er beskrevet i et eget dokument og er således ikke et fokus her, men typiske utfordringer med en kravspesifikasjon sett i lys av datadesignet er:

- Kravspesifikasjonen har funksjonelt fokus, for eksempel i form av en rapportspesifikasjon (felter i rapporten, menyvalg etc.), mens dataunderlaget til rapporten eller funksjonen ikke er like godt definert.
- Kravspesifikasjonen er avgrenset til et bestemt formål og brukermiljø, mens datadesignet bør være mest mulig generisk og ta høyde for øvrige eller fremtidige behov knyttet til de samme dataene.

Dette betyr ikke at kravspesifikasjonen har feil fokus. Tvert i mot; det er prosesstøtten som gir forretningsnytte, ikke dataene i seg selv. Dataene kan ansees som infrastruktur, og det alene gir gjerne en dårlig business case. Det er for øvrig også riktig å avgrense prosjektene så mye som mulig for å sikre håndterlige omfang og korte iterasjoner, blant annet for å unngå at behovene har endret seg vesentlig før prosjektet er ferdig.

Funksjonelle krav i spesifikasjonen kan likevel peke direkte på databehovet, for eksempel krav til oppdateringsfrekvens og hvilke historiske endringer som skal fanges opp i dataene.

Dataarkitektens rolle er å styre prosessen for å avdekke det underliggende databehovet i kjølvannet av den funksjonelle spesifikasjonen, og å designe datamodellene. Denne prosessen kan være krevende, både fordi det kan være vanskelig å finne konsensus når alle de forskjellige datainteressentene kommer sammen, men også fordi brukerne som har bestilt prosjektet ofte har en ren funksjonell tilnærming til problemstillingene. Dataarkitekten må sørge for å konsentrere diskusjonene rundt data.

## Adopter brukernes begrepsapparat i navn og definisjoner

I BI-prosjekter vil vi ofte ende opp med en stor gruppe interessenter, og vil typisk møte problemstillingen med ulik bruk av begreper i de forskjellige subkulturene. Dataarkitektens rolle er å fange opp essensen og styre mot mest mulig omforente begreper. Bedriftskulturen er avgjørende for hvor formelle disse prosessene og beslutningene bør være. I slike prosesser er det nyttig å samle brukerne på tvers av disse subkulturene i workshops for å identifisere og synliggjøre avvikene. Ofte må vi gjennom flere iterasjoner inntil en felles konsensus vokser frem.

På et tidlig stadium kan brown paper session være en god metode for å kartlegge og kategorisere dataene. Dette er en prosess som starter med at alle deltagere noterer sine begreper på post-it lapper og deretter grupperer disse på store papirark på veggen (herav navnet brown paper). Etter hvert som lappene plasseres og grupperes, vil begrepsavvik og overlapp bli tydeligere, og gode diskusjoner rundt definisjoner vil oppstå.

Senere i prosessen, i diskusjoner om sammenhengen mellom dataene, vil whiteboard bli et viktig verktøy. Da har dataarkitekten en sentral rolle i diskusjonene. Brukerne vil stadig spore diskusjonene inn på prosesser og funksjoner, og da er det viktig å styre diskusjonen tilbake til databehovet. Kartlegging av forretningsprosesser er absolutt en viktig del av et BI-prosjekt, men for å sikre data usability er det nødvendig å holde på datadesignet som en rendyrket aktivitet. Å knytte data til forretningsprosesser gjøres senere.

På dette stadiet spør vi hvilke data, detaljer og egenskaper er det viktig å ta vare på innenfor subjektområdet vi diskuterer – hva er verdt å huske?

Navngivningen er sentral for å sikre god brukskvalitet. På dette stadiet i prosessen utvikles den logiske modellen. Da tas det ikke hensyn til tekniske begrensninger i databaseapplikasjonen, for eksempel restriksjoner på tegnsett, maksimal lengde på navn etc. Dette tas hensyn til i fysisk modell. Vi fraråder bruk av tekniske strukturer og forkortelser. Bruk hele ord og mellomrom. Standardiser gjerne på typiske begreper, for eksempel identitet, kode, flagg, osv.

Alle navn skal være:

- Selvforklarende: Det er ingen grunn til å bruke kryptiske eller korte navn.
- Presise: Gi et navn som eksakt beskriver innholdet. Unngå tvetydighet.
- Unike: Det skal ikke kunne forveksles eller sammenblandes med andre data.
- Standardiserte: Dersom det finnes mange synonymer, velg ett av dem og standardiser på dette.

Absolutt alle dataelementer, både entiteter og attributter, skal ha tydelige definisjoner og beskrivelser. Ingen beskrivelsesfelt skal stå åpne. Gode beskrivelser følger de samme reglene som for navngiving. I tillegg bør det defineres:

- Datatype, for eksempel tekst, nummer, dato etc.
- Gyldige verdier, særegenskaper, intervaller og grenseverdier
- Datakilde – hvor dataene kommer fra
- Oppdateringsfrekvens – for eksempel daglig eller månedlig
- Datakvalitet – har dataene blitt kvalitetssikret, ugyldige verdier fjernet, etc.
- Tilgangsrestriksjoner i de tilfellene hvor dataene er av sensitiv art

Vi kan ende opp med begrepskonflikter som går i vranglås. I slike tilfeller må vi la de ulike subkulturene beholde sine begreper. Det naturlige er å løse dette ved å benytte forskjellige begreper enten gjennom semantiske lag som døper om dataene i brukergrensesnittet, eller ved å etablere adskilte datamarter for de ulike miljøene. I datavarehuslaget må det velges kun ett av begrepene. I slike tilfeller er det de som bruker BI-løsningen fra et overordnet virksomhetsperspektiv som bør ha siste ordet. Alternativt må sjefsarkitekten skjære igjennom og ta et valg. Sammenhengen mellom data i datamart og datavarehus må dokumenteres. Vi må sikre at begrepet som brukes på tvers er forstått, selv om en subkultur velger å holde på sitt.

Sporbarhet tilbake til dataenes kilde er sentral informasjon. Det finnes også regulatoriske krav til data-sporbarhet innen en del markeder og bransjer, for eksempel SOX, Basel og Solvency. Dataelementenes

kilder kan dokumenteres på forskjellige måter. Bruk av et såkalt mappingdokument er en vanlig spesifikasjonsmetode som gjerne inngår i detaljdesignet. Noen ER-modelleringsverktøy (entity-relationship model), for eksempel CA ERwin Data Modeler, har funksjonalitet for registrering av datakilde som en del av egenskapsdefinisjonen på hvert enkelt attributt. Dataintegrasjonsverktøy har gjerne funksjonalitet for sporbarhetsrapportering (ofte kalt lineage eller impact analysis). Denne funksjonaliteten er dessverre fortsatt lite tilgjengelig i de fleste sluttbrukerverktøy.

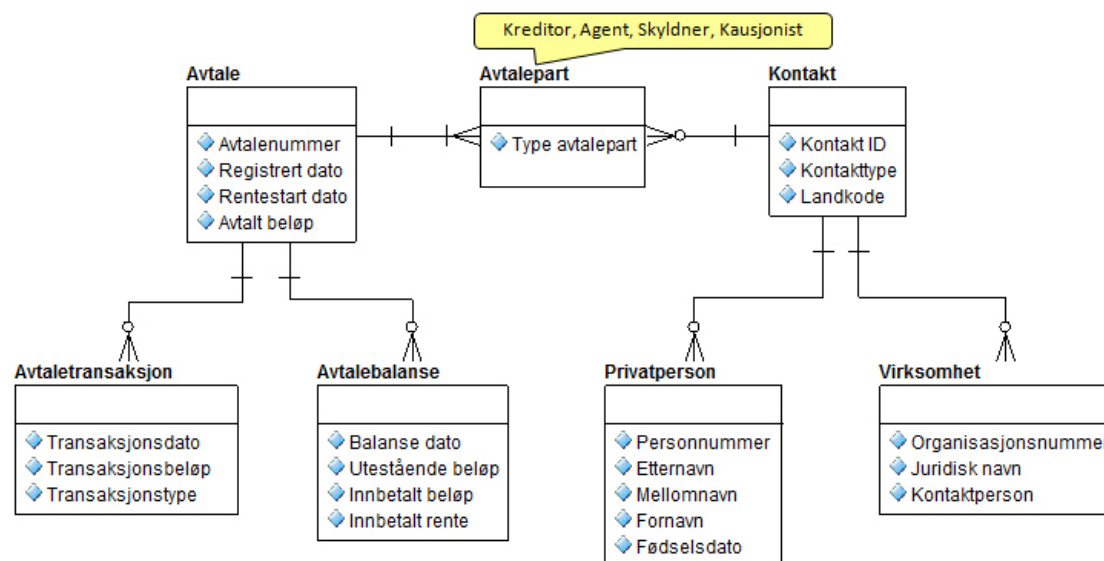
## Involver brukerne i datamodelleringen

Datamodellering er ikke en IT-teknisk aktivitet. Definisjon av relasjonene mellom dataene krever samme brukerinvolvering som datadefinisjonene, og er en naturlig del av datadefinisjonsprosessen. Det er i utgangspunktet ikke intuitivt for brukere å skille mellom dataenes attributter og relasjoner, så dette krever en viss pedagogisk innsats innledningsvis, men det er ikke vanskelig å involvere brukerne i datamodelleringen.

En logisk datamodell er enkel å forstå og et ekstremt viktig verktøy i interaksjonen mellom forretning og IT. Med enkle teknikker kan vi introdusere symbolene for brukerne og raskt komme inn i gode dialoger om dataenes logiske sammenheng. En god teknikk for å bygge opp brukernes forståelse er at dataarkitekten tegner opp på whiteboard etter hvert som dataenes egenskaper kommer opp i diskusjonen. Her er et illustrert eksempel:

- Ta utgangspunkt i en sentral entitet (tabell), for eksempel Avtale
- Skriv opp de viktigste attributtene (variablene) til avtalen, for eksempel Avtalenummer og Dato
- Involver brukerne – be om flere attributter.
- På et tidspunkt vil det komme opp et forslag som krever en relasjon, for eksempel Avtalepart
- Det viser seg at alle avtaleparter ligger registrert i et kontaktregister. Tegn opp entiteten Kontakt
- Tegn en midlertidig strek mellom Avtale og Kontakt
- Få opp hvilke typer avtaleparter som finnes, for eksempel Kreditor, Agent, Skyldner og Kausjonist
- Skriv opp disse over streken. Diskuter det faktum at antall avtaleparter kan variere fra avtale til avtale
- Ta bort streken og sett inn Avtalepart med strek mellom Avtale, Avtalepart og Kontakt
- Tegn relasjonsnotasjoner (kråketær) på linjene på hver side av Avtalepart samtidig som du forklarer at:
  - en avtale må ha en eller flere avtaleparter
  - en avtalepart må eksistere i kontaktregisteret
  - en kontakt kan være avtalepart til flere avtaler

Det skal ikke mange bokser og relasjoner til før brukerne har forstått notasjonene og logikken. Kanskje første modell blir som eksemplet under:

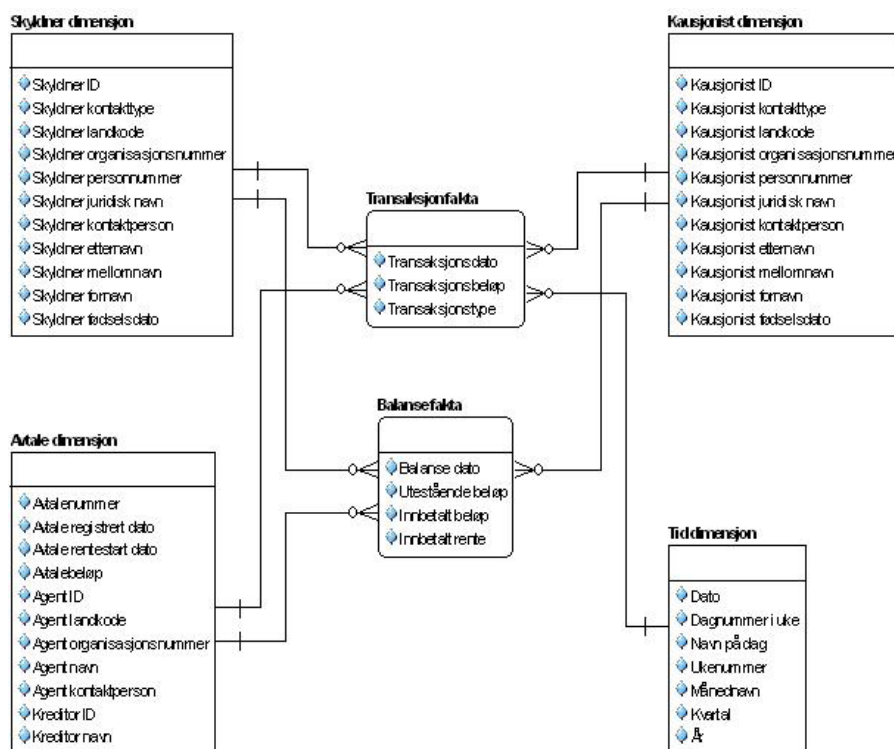


Det viktige er at det tegnes opp boks for boks. Dersom vi kaster opp en full modell, sikrer vi verken forståelse eller brukerinvolvering. Det er først når brukerne engasjerer seg i det som blir tegnet opp, at de konstruktive innspillene kommer og at brukerne tar eierskap til datamodellen. Etter hvert blir situasjonen snudd, og det er dataarkitekten som lærer av brukernes forretningsforståelse.

Start gjerne hver sesjon med blank whiteboard, selv om møtet er en fortsettelse fra forrige gang. Det går raskt å tegne opp entitetene på nytt, og det er ikke nødvendig å ta med alle attributtene hver gang. Resultatet fra forrige workshop skal selvsagt foreligge dokumentert og detaljert, fortrinnsvis i et ER-modelleringsverktøy. Hver gang dataarkitekten tegner modellen på whiteboard, økes både egen forståelse og brukernes forståelse, og nye momenter kommer gjerne opp.

En typisk fallgrube er at brukerne har et forenklet begrepsapparat. I eksemplet over kan det være at begrepene transaksjon og balanse er de som blir brukt. Dette er greit innenfor et bruksområde, men disse begrepene er alt for generelle for en virksomhetsomspennende BI-løsning. I dette tilfelle dreier det seg om transaksjoner og balanse knyttet til avtalen. Dersom brukere fra økonomiavdelingen hadde vært involvert, ville de antagelig brukt samme begrepene om hovedboktransaksjoner og -balanse. Derfor er det viktig å legge på denne konteksten i navnene. Applikasjonsnavn kan også ha blitt innarbeidet i begrepene, for eksempel Agressotransaksjon. Selv om Agresso er økonomisystemet for øyeblikket, er det svært viktig å unngå begreper som inneholder navnet på en IT-applikasjon. Da må vi ta utgangspunkt i applikasjonens funksjon og finne riktig begrep ut fra det, for eksempel regnskapstransaksjon.

Datamartdesign er i større grad knyttet til funksjonelle krav. Mens vi i datavarehusdesignet holder fokuset på hvilke data det er viktig å ta vare på og hvordan henger de sammen, vil vi i datamartdesignet typisk spørre hvilke faktatall skal vi måle og hvilke dimensjoner skal vi knytte tallene til. Da vil kanskje den logiske modellen bli som eksemplet under:



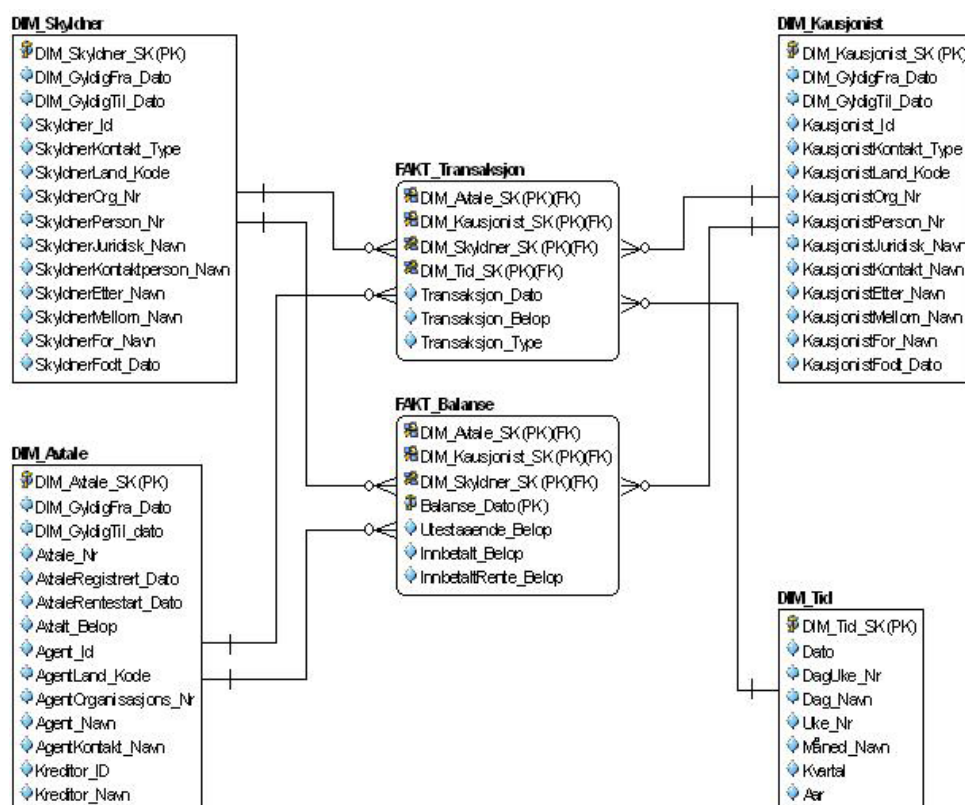
I dette tilfellet har vi laget en modell tilpasset en spesifikk brukergruppe, og da kan vi tillate lokale begreper som transaksjon og balanse. Her har avtalepartene Skyldner og Kausjonist blitt tydeligere i form av egne entiteter (de kan likevel være alias eller view i fysisk modell), mens Agent og Kreditor bare er lagt inn som attributter på Avtale. Dette er en modell som er bedre tilpasset dynamisk rapportering rundt transaksjonene og balansen, men for annen bruk er den ikke like god. For å se hvilke skyldnere og kausjonister som er knyttet til en avtale, må vi i denne modellen gå via for eksempel balansen. Det kan være greit for denne brukergruppen, men kanskje upraktisk for andre.

## Litt om fysisk design og implementering

Den fysiske datamodellen utledes i utgangspunktet direkte av den logiske modellen. Dette gjelder spesielt for datavarehusdesign, da beste praksis innen datavarehusutvikling sier at datamodellen skal være subjektorientert. I praksis betyr dette at tabeller skal navngis slik at de tydelig angir innholdet og at universalta-beller bør unngås. Dette er som regel ikke tilfelle for de operasjonelle systemene som er kildene til data-varehuset. Mange standardsystemer har utstrakt bruk av universale tabeller, med generelle navn, som kan inneholde forskjellige typer data avhengig av kundetilpassede parametere. En god datavarehusmodell er av den grunn langt mer lesbar for ikke-tekniske brukere enn datamodellene til operasjonelle systemer.

Den fysiske modellen vil likevel bli noe annerledes enn den logiske. Mens vi i den logiske modellen kan bruke norske tegnsatt, spesialtegn og mellomrom for å øke lesbarheten, må den fysiske modellen ta hensyn til den valgte databaseapplikasjonen. Det kan også være andre tekniske betraktninger rundt for eksempel datamengder, endringstakt og ytelse som gjør at de fysiske tabellene kan bli noe annerledes enn de logiske entitetene. Dette er årsaken til å holde logisk og fysisk datamodellering som to adskilte aktiviteter. Brukerne skal ikke involveres i sistnevnte.

Typiske BI-verktøy, for eksempel rapporteringsverktøy, tilbyr semantisk lag for å skjerme brukeren for komplekse datastrukturer, blant annet gjennom bedre navngiving og beskrivelser. Dette bør likevel ikke forhindre bruk av de samme retningslinjene for god navngiving i den fysiske modellen. Det er ingen grunn til å skape ekstra kompleksitet. Hva er for eksempel god usability for analytikere? En typisk analytiker ønsker gjerne fri tilgang til dataene og vil ikke akseptere å bli låst til en enkel applikasjon med sine begrensninger. Da nytter det ikke å skjerme dårlige datanavn bak et semantisk lag. Det vil alltid være brukere som har behov for direkte tilgang til dataene. Vi skal heller ikke glemme utviklere og andre IT-ressurser som har behov for den samme dataforståelsen som brukerne. Under ser vi et eksempel på fysisk modell:



Modellen inneholder tekniske kolonner som surrogatnøkler og tidsintervall for versjonshåndtering av dataene. For tydelig å skille ut de attributtene som kun finnes i den fysiske modellen, har disse fått DIM\_ som prefiks i dette eksemplet. De tekniske kolonnene er ikke normalt å ta med i logiske modeller. Det skaper unødvendig kompleksitet for brukerne å se attributter som: Surrogatnøkkel, tidsstempel, lastejobb ID, sporingskoder etc.

Bruk av navnekonvensjoner er en god måte å øke lesbarheten:

- Bruk prefiks og suffiks til å indikere hva slags type kolonnen inneholder: `_Id`, `_Navn`, `_Kode`, etc.
- Standardiser oppbyggingen av navn, for eksempel skille ord med stor forbokstav, rolle først, type sist etc.
- Norske tegn (æ,ø,å) skaper ofte problemer i fysisk implementasjon. Definer tydelig hva de erstattes av.
- Bruk konsernspråket (som regel engelsk) dersom virksomheten er etablert i flere land

Et godt ER-modelleringsverktøy gir gjerne muligheten til å bygge logisk og fysiske modell parallelt. Det er for øvrig delte meninger om det er riktig prosessuelt å gjøre dette. Motargumentet går ut på at man ikke skal tenke teknisk under design av logiske datamodeller. Det er for så vidt riktig, men det er likevel en god og effektiv rutine å definere fysiske navn og datatyper samtidig som vi legger inn de logiske elementene. Det er svært effektivt å definere et ferdig sett med datatyper, såkalte domener, i forkant av modelleringsarbeidet. Mange verktøy gir mulighet for etablering av makroer vi kan sette opp til å følge navnekonvensjonene.

Datadefinisjonene må være med inn i fysisk design og teknisk løsning. Det hjelper lite om datadefinisjonene havner i et bortgjemt dokument. Definisjonene må være tilgjengelige i de grensesnittene brukerne jobber, for eksempel i analyseverktøyet eller webportalen. De må også være tilgjengelige for utviklere, for eksempel utviklere av dataintegrasjonsjobber. Alle datadefinisjoner bør være tilgjengelige i ETL-verktøyet.

Datadefinisjonene skal registreres på alle elementer i ER-modelleringsverktøyet. Gode verktøy har funksjonalitet for oversikt, rapportering og analyse på alle metadata. Det viktigste er dog gode eksportmuligheter. Definisjonene må overføres til:

- Den fysiske databasen (for eksempel Extended Attributes i Microsoft SQL Server)
- Metadata Repository i ETL/dataintegrasjonsverktøyet
- Semantisk lag i analyse- og rapporteringsverktøy (for eksempel SAP BO Univers)

## Oppsummering

Business intelligence dreier seg om anvendelse av data til multiple formål. Brukergrensesnittet er kun en innpakning av dataene. Det er dataene som er hovedproduktet. Selv om de fleste BI-prosjekter avgrenses av et sett med definerte funksjonelle krav, som for eksempel gjennom rapportspesifikasjoner eller analysefunksjoner, vil disse gjerne endre seg over tid. Stadig nye innfallsvinkler og analyserbehov krever at datadesignet må sees i en større sammenheng enn enkeltprosjektets kravspesifikasjon. Dataarkitekten må fokusere på å få datadesignet FIKKSST ferdig:

- **Fleksibelt:** Er dataene godt tilrettelagt for analyse og ad-hoc rapportering?
- **Intuitivt:** Formidler dataenes navn og beskrivelse en effektiv og intuitiv forståelse av innholdet?
- **Konsistent:** Er dataene fullstendige og entydige på tvers av virksomheten?
- **Kvalitativt:** Er datakvaliteten kjent? Er kravene definert? Er dataene vasket/beriket?
- **Skalerbart:** Hvor enkelt det er å sammenstille dataene på andre måter til nye bruksområder?
- **Sporbart:** Hvor kommer dataene fra? Hvem produserer og vedlikeholder dem?
- **Tilgjengelig:** Er dataene tilgjengelige for brukerne, der de trenger den, når de trenger dem?

## Referanser

- Building Enterprise Information Architecture, Melissa A. Cook
- Building the Data Warehouse, William H. Inmon
- Improving Data Warehouse and Business Information Quality, Larry P. English
- Mastering Data Modeling: A User Driven Approach, John Carlis, Joseph Maguire
- The Data Warehouse Lifecycle Toolkit, Ralph Kimball, Margy Ross, Warren Thornthwaite, Joy Mundy, Bob Becker

Affecto er den ledende Business Intelligence-leverandøren i Norden, med mer enn 900 medarbeidere i Finland, Norge, Sverige, Danmark, Baltikum og Polen.

Affecto Norway  
GrevWedelsplass5  
0151 Oslo  
tel: +47 22 40 20 00  
info@affecto.com  
www.affecto.no