**Database modeling notes**

**Week: 1 / Topic: Data modeling / PowerPoint: Anniken /Språk: Norsk**

**Data modellering**

**DBMS**

Et database håndteringssytem (DBMS) er et datastyrt system som lar brukere lage og opprettholde en database. DMBS er et programvaresystem (software) som legger til rett for:

* Definering = spesifisere datatyper, strukturer og begrensninger til dataen
* Konstruksjon = lagre dataen på et medium som kontrolleres av DBMS
* Manipulasjon = henting av spesifikk data, oppdatering pga. endring i miniverden, osv.
* Deling = lar flere brukere og programmer få tilgang til databasen samtidig

**Databasesystemet**

* Programmer får tilgang til databasen ved å sende forespørsler om data til DBMS.
* Ved en transaksjon kan noe data bli lest eller skrevet inn i databasen.
* Andre viktige funksjoner ved DBMS inkluderer beskyttelse av database og vedlikehold/bruk over lengre tid (oppnås vha tuning).
* Beskyttelse innebærer systembeskyttelse fra funksjonsfeil og sikkerhet mot uautorisert eller skadelig tilgang.
* DBMS og databasen kombinert kalles databasesystemet.
* Databasesystemet inneholder en DBMS katalog som lagrer beskrivelsen av databasen (eks: datastruktur, typer og begrensninger). Denne beskrivelsen kalles metadata.

**Deling av data og flerbrukets transaksjonsbehandling**

* En flerbruker DBMS må la flere brukere få tilgang til databasen samtidig.
* DBMS må ha programvare fro samtidighetskontroll, slik at når flere brukere prøver å oppdatere den samme dataen, så blir dette gjort på en kontrollert og effektiv måte.
* For eksempel ved salg av flyseter må DBMS sikre at kun en bruker får reservere ett bestemt sete. Dette kalles OLTP (Online Transaksjon Prosessering) og lar flere hundre samtidige transaksjoner utføres per sekund.
* Transaksjoner er utførende programmer som inkluderer akses til en eller flere databaser, for eksempel lesing eller oppdatering av databaseeposter.
* Samtidighetskontroll garanterer at hver transaksjon blir riktig utført eller avbrutt.
* DBMS må sikre flere egenskaper ved transaksjoner, for eksempel isolasjon (hver transaksjon ser ut til å utføres isolert fra andre som utføres samtidig) og atomisitet (alle operasjonene i en transaksjon blir utført eller ingen).

**I Store Organisasjoner vil mange personer være involvert i designet og opprettholdelsen av større database**

* Database administratorer (DBA) – styrer databasen, DBMS og relatert programvare som blir brukt av mange personer.
* DBA autoriserer aksess til databasen, koordinerer bruken, henter program- og maskinvare som trengs og sørger for sikkerhet og god responstid.
* Database designere – identifiserer dataen som skal lagres i databasen og velger passende strukturer for å representere og lagre dataen.
* Sluttbrukere – brukerne som er grunnen til at databasen eksisterer.
* System analytikere og applikasjonsprogrammerere – utvikler programvaren og bestemmer kravene til brukerne, tester, debugger, osv.
* Andre personer er ansvarlig for design, utvikling og drift av DBMS programvaren og systemmiljøet. Disse er som regel ikke interessert i selve databaseinneholdet: DBMS systemdesignere, verktøyutviklere, personell for drift og vedlikehold.

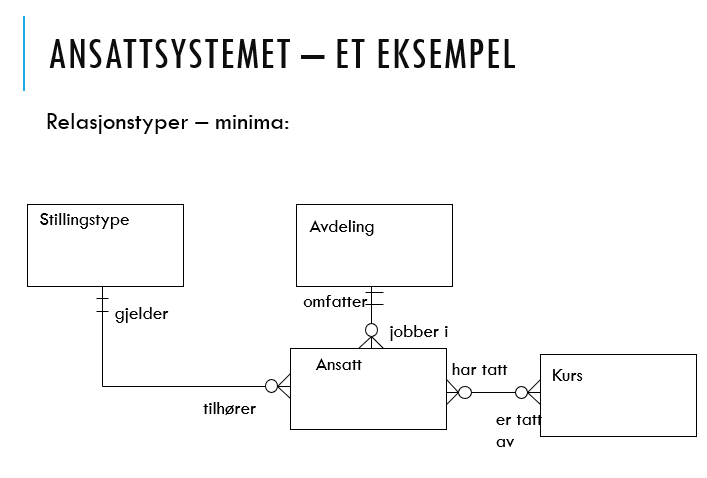
**Fordeler ved å bruke en DBMS**

* Kontroll av redundans
* Hver bruker opprettholde sine egne filer, noe som fører til duplikater av samme fil.
* Gir bortkastet lagringsplass og mer arbeid for å oppdatere dataen.
* DBMS bruker redundansskontroll for å sikre at disse duplikatene er riktige.
* Begrenser uautorisert tilgang
* Gir vedvarende lagring for programobjekter – et kompleks objekt skrevet i C++/Java kan lagres permanent i en objektorientert DBMS, slik at den «overlever» terminering av programmet og kan senere direkte hentes av et annet program.
* Gir lagringsstruktur og søketeknikker for effektiv forespørsel.
* Gir backup og gjenoppretting – DBMS legger til rette for gjenoppretting etter feil i maskinvare eller programvare.
* Gir grensesnitt for flere brukere.

**Entitetstype**

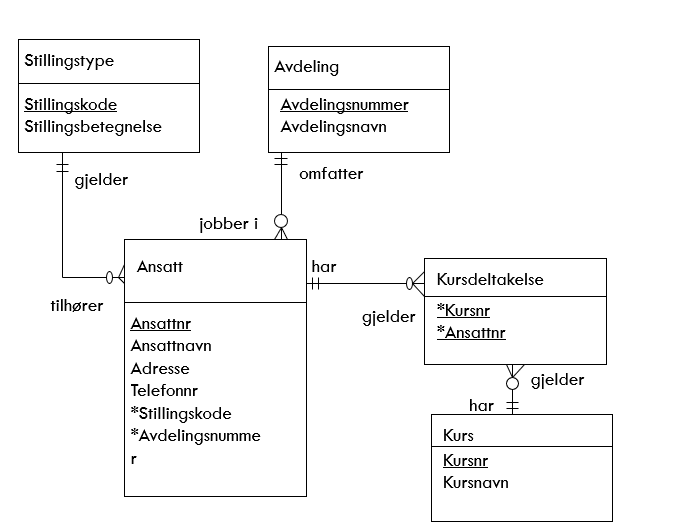
**Entitet**: Et eller annet innenfor vårt interesseområde som vi ønsker informasjon om.

**Entitetstype**: En samling entiteter der alle entitetene oppfyller definisjonskriteriene for entitetstypen.



**Hoved- og fremmednøkler**

* For at datamodellen skal kunne brukes som grunnlag for en database må basen ha en form for «kobling» mellom de ulike entitetstypene.
* Vi må bringe inn hovednøkkel og fremmednøkkel-begrepet.
* Hovednøkkelen – unik.



**Identifikator = Primærnøkkel = Hovednøkkel**

* Vanligvis består identifikatoren av ett eneste attributt
* De forekommer identifikatorer som er sammensatt av to eller flere attributter = sammensatt identifikator

**Identifikator – Definisjon**

En indentifikator for en enititetstype er et attributt eller en gruppe av attributter som velges ut til å identifisere de ulike forekomster av en entitetstype, og som oppfyller følgende egenskaper: Entydighet, irredusibilitet og entitetsintegritet.

**Entydighet** = Alle verdir for attributtet er forskjellige

**Irredusibilitet**: Aktuelt ved sammensatt identifikator: Skal ikke kunne kutte ut ett eller flere attributter og likevel kunne bruke de gjenværende som identifikator

**Entitetsintegritet**: Identifikatoren må ha en verdi.

**Week: 2 / Topic: Module 2 – Data modeling / PowerPoint /Språk: Engelsk**

**Week: 2 / Topic: Module 3 – SQL / PowerPoint /Språk: Engelsk**

**Week: 2 / Topic: Module 14 Database security / PowerPoint /Språk: Engelsk**