

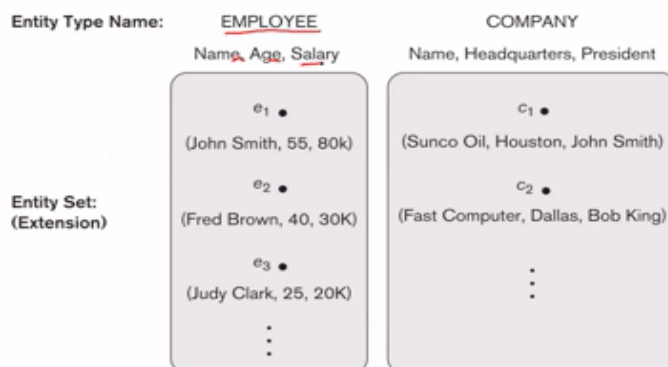
Entity-Relationship (ER) Modell

- Definerer en datastruktur som kan bli implementert av en database
- En (grafisk) språk som spesifiserer:
 - Informasjon databasen skal inneholder
 - relasjoner blant komponentene av den informasjonen
- Presist nok til å støtte implementasjon men abstrakt for non-database folk
- Designs er bilder kalt entity-relationship diagrams

Entities

- **Entity:** "Ting" eller objekter. En abstrakt representasjon av et objekt/ting som kan være knyttet til den virkelige verden, f.eks John(Navn), IN2090(Fag).
- **Attribute:** Egenskaper som beskriver en entitet. En entitet kan ha flere attributter som navn, alder, fødselsdato osv. For hver entitet så vil hver attributt ha en verdi. Entitet -> Navn, Attributt -> John, Entitet -> Alder, Attributt -> 23
- **Entity type:** Samling av entiteter som deler de samme attributtene. ANSATT har Navn og Alder f.eks. Entiteter som rett og slett består av flere entiteter sammen.

Entity sets



Entity types in ER diagrams

- An entity type is depicted as a *rectangle* consisting the entity type name
- An attribute is depicted as an *oval* consisting the attribute name, with a line to the rectangle representing its entity type



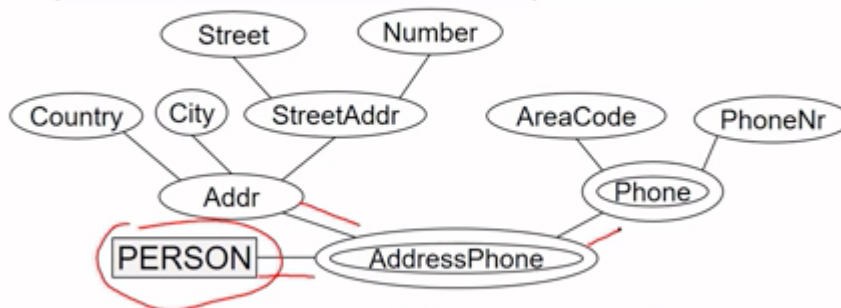
- Entity type EMPLOYEE has three attributes, EID, Name, and Bdate
- Each EMPLOYEE entity has values for these three attributes, e.g. (1, John, 19-09-1999)

Types of attributes

- **Composite vs Simple**
 - Composite: attribute that is divided into smaller subparts, possibly forming a hierarchy
 - Simple: attribute that is not divisible
- **Single-valued vs Multi-valued**
 - Single-valued: attribute that has a single value for a particular entity, e.g.,
John's Age is 23
 - Multi-valued: attribute that can have a set of values for the same entity, e.g.,
John's University_degrees are BSc and MSc
- **Stored vs Derived**
 - Derived: attribute whose value can be determined from the value of other related attributes, e.g.,
John's Age can be determined from the current date and the value of John's Birth_date
 - Stored: attribute whose value is used to determine the value of a derived attribute

Complex attributes

- Complex attributes: nested composite and multi-valued attributes, e.g.,
A person with more than one address and phone number



- Group components of a composite attribute between parentheses (), separate the components with comma, and display multivalued attributes between braces {}
{AddressPhone((Phone(AreaCode, PhoneNr)), Addr(StreetAddr(Number, Street), City, Contry))}

Attribute values

- **Null value**
 - Samme tankegang som ved koding når vi har en nullpeker, hvis en entitet ikke har en verdi for en attributt, så vil den få tilegnet verdien NULL
 - Ukjente tilfeller: At John's telefonnummer er null, som kan skyldes av at John ikke har et telefonnummer
 - Ikke tilgjengelig: Om John ikke har en universitetsgrad, så vil det stå John's universitetsgrad er NULL
- **Value sets**
 - Beskriver hvilke attributter som er tillatt av den gitte datatypen. F.eks så vil domene (Studen) kun tillatte alder av integer(tall) numre mellom 16 til 65.

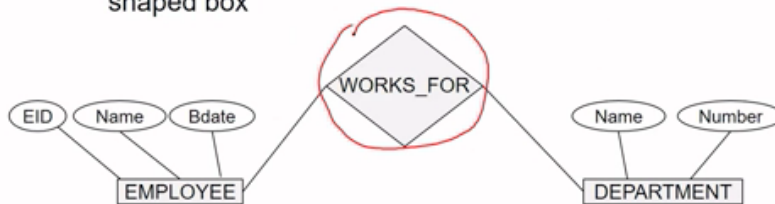
Begreper knyttet til Relationships

- **Relationship:** En assosiasjon mellom to eller flere entiteter
- **Relationship type:** En samling av lignende assosiasjoner, f.eks JOBBER_FOR en relasjonstype mellom ANSATT og DEPARTEMENT
- **Degree of a relationship type:** Et nummer som beskriver antallet av entity types.

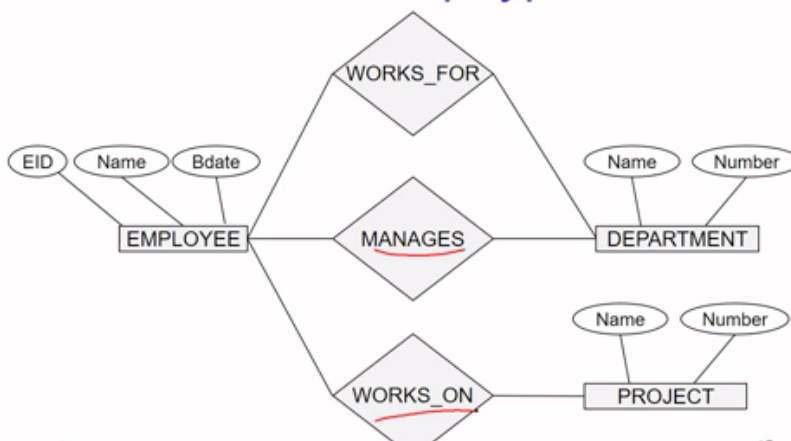
- Binary: Relationship type of degree two, the WORKS_FOR has degree two
- Ternary: Relationship type of degree three
- N-ary: Relationship type of degree N

Relationship types in ER diagrams

- Represented as diamond-shaped boxes, connected by straight lines to the rectangular boxes representing the participating entity types
 - The relationship name is displayed in the diamond-shaped box



ER diagram example entity and relationship types



Constraints in ER

Constraint: En påstand om at databasen alltid skal være sann. Alt vi gjør stemmer overens med Entity-relationship databasen, vi skal unngå å måtte registrere attributter til uriktige entiteter, godkjenne effektiv lagring, tillate oss å referere til entiteter osv.

Key constraint: Hver entitet har en attribut som er unikt å identifisere.

Participation constraint: Spesifiserer hvorvidt en entitet er gyldig avhengig av om den er relatert til en annen entitet gjennom en relationship type.

- Total participation: Hver ansatt må jobbe for et departement. En ansatt entity kan bare eksistere bare hvis den er en del av minst en JOBBER_FOR relationship instance.
- Partial participation: Noen ansatte er relatert til noen departement entity gjennom MANAGES, men ikke nødvendig i det hele tatt.

Participation constraints in ER diagrams

- Total participation is displayed as a double line connecting the participating entity type to the relationship
- Partial participation is represented by a single line



Key constraints

- An important constraint on the entities of an entity type, a.k.a. **uniqueness constraint** on attributes
- **Key attribute**: an attribute whose values are distinct for each individual entity in the entity set (i.e., its values can be used to identify each entity uniquely), e.g., *The EID attribute of EMPLOYEE is a key because no employees are allowed to have the same id*
- In ER diagrams, each key attribute has its name underlined inside the oval

