

# 732G16: Databaser: Design och programmering

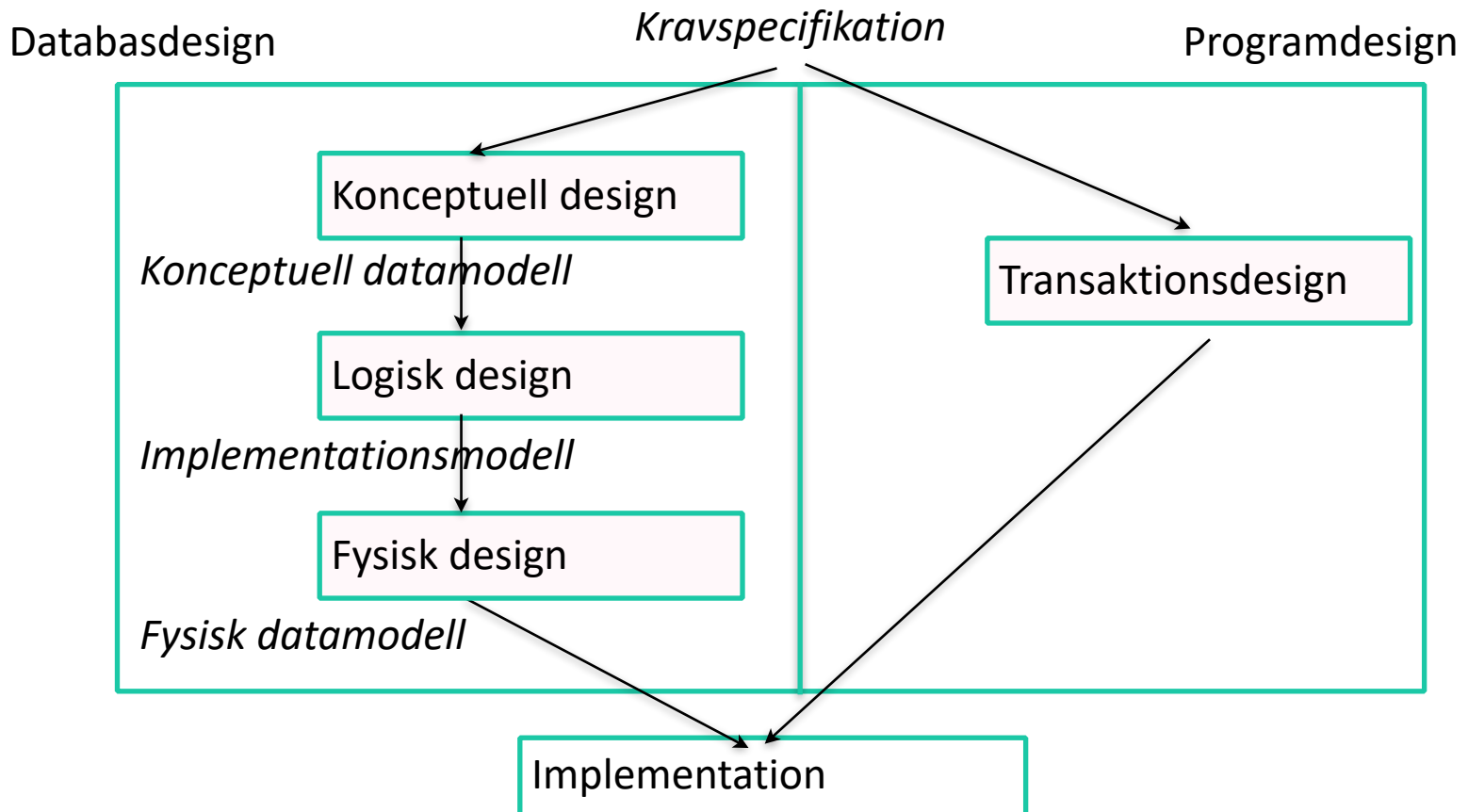
Fö 3: Relationsmodellen

2026

# Innehåll:

- Relationsmodellen - definition
- Konvertering från ER till relationer

# Programdesign, databasdesign



# Relationsmodellen

[www.liu.se](http://www.liu.se)

# Relationsmodellen

- Introducerades av Edward Codd 1970
- Mycket vanlig
- Stödjer kraftfulla och ändå enkla deklarativa språk
- Matematisk grund i relationsalgebra

# Algebra: Funktioner

- Funktion = avbildning (mappning)

$$Y=X^2$$

- Definitionsmängden mappas till värdemängden
- Kan representeras med tabell
- Varje värde i definitionsmängden mappas på ett värde i värdemängden (men ej omvänt)

# Relation = generaliserad funktion

- Tabellen har fler än två kolumner - mer än två mängder kopplas ihop med varandra.
- Värden på samma rad ”hör ihop”
- Mängden som värden dras från: domän

# Formell definition av relation, R

- En relation är en delmängd av kartesiska produkten av ett antal domäner A, B, C...
- Formellt:  $R \subseteq (((A \times B) \times C) \times \dots)$
  
- $\times$  Kartesisk produkt: Mängden av alla ordnade par av element från två domäner

# Exempel

- om  $D1 = \{\text{Anna, Olle}\}$  och  $D2 = \{\text{student, anställd}\}$
- så blir  $D1 \times D2 = \{\langle \text{Anna, student} \rangle, \langle \text{Anna, anställd} \rangle, \langle \text{Olle, student} \rangle, \langle \text{Olle, anställd} \rangle\}$
- Relationen Person kan då t.ex vara den verklighetsbaserade delmängden  $\{\langle \text{Anna, student} \rangle, \langle \text{Olle, anställd} \rangle\}$

# Relationer - som tabeller

Relationen har **attribut** och är en mängd av **tupler**

Attribut

Attributvärde

Tupel

<b>Student</b>		
Pnr	Namn	roll
801212	Anna B	anställd
950101	Oskar A	student

# Relationer – som tabeller forts.

- **tupel** = **ordnad** lista av attributvärden.  
<9010111234, Olle, anställd>
- relation = mängd av tupler (ingen ordning, inga dubletter)
- antalet attribut kallas tupelns **grad**.
- attributvärdena är **atomära** (odelbara).

# Relationsschema

Formell beskrivning av viss relation:

relationsnamn (attributnamn1, attributnamn2, ...  
attributnamnN).

- Exempel:
  - Person (Namn, Adress, Telefon, Födelsedatum, position)
- Namnet på attributet antas beskriva domänen.

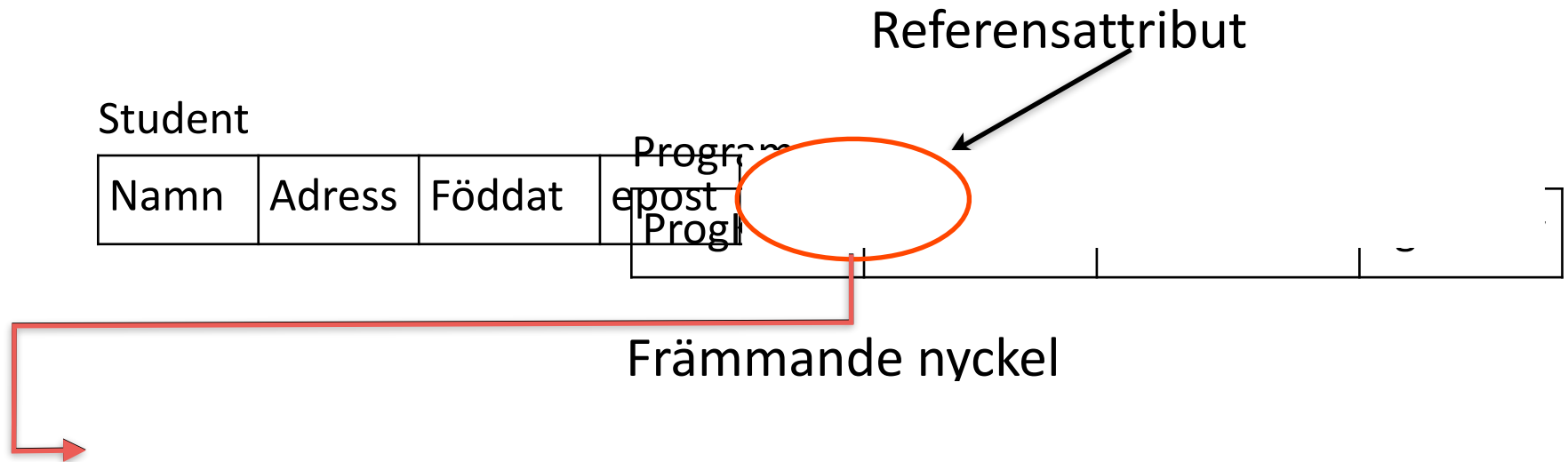
# Alternativa notationer:

Person

Namn	Adress	Föddat	Kön	Tel
------	--------	--------	-----	-----

Person
Namn
Adress
Föddat
Kön
Tel

# Mer information?



- En databas består av flera relationer med kopplingar mellan sig.

# Referensintegritet

- Delete from Program where Pgmkod="f7ksa"
  - Delete from Program where Pgmkod="6cddd"
- Student

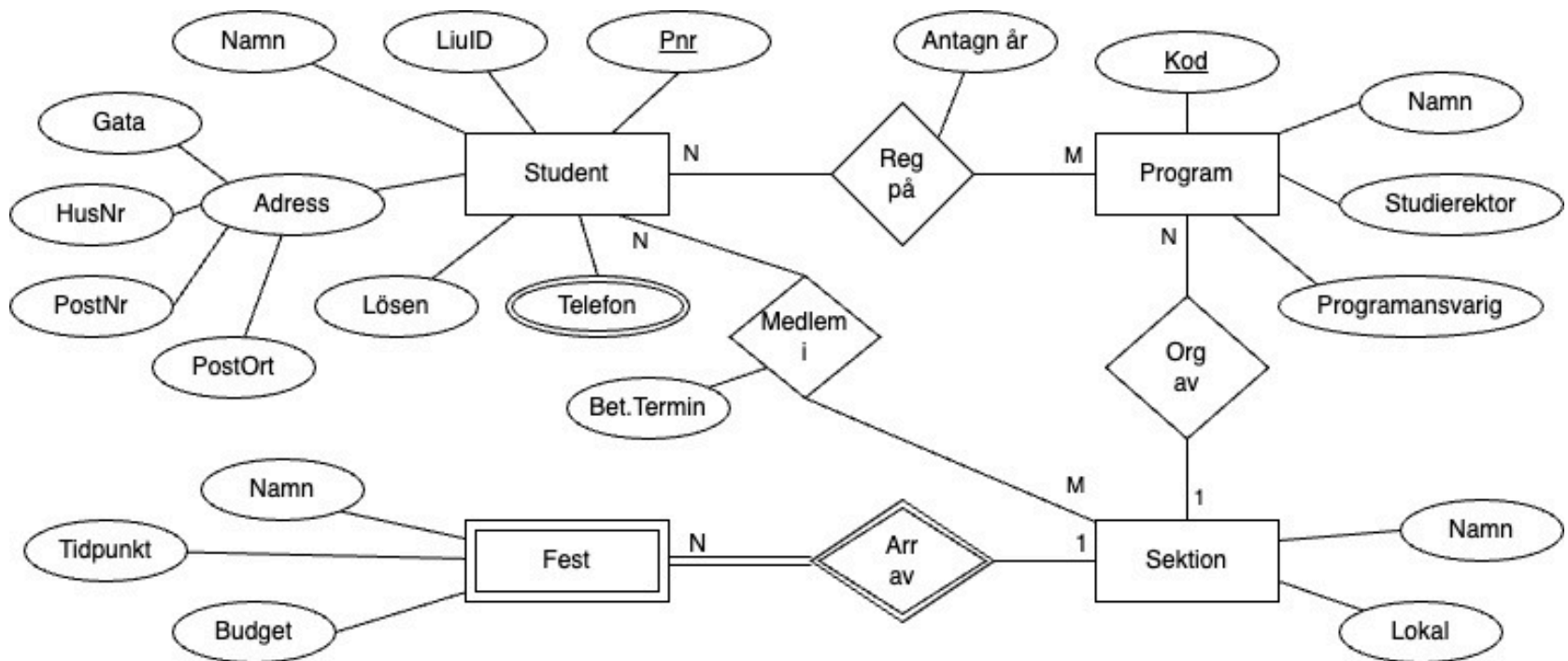
Namn	Adress	Föddat	epost	Program
Pelle	Bygat 2	970715	p@gm	f7ksa
Anna	Rydsv 3	971224	<u>a@liu.se</u>	f7kko

## Program

Pgmkod	Namn	Studierektor	Programansvarig
f7ksa	Statistik o data	Lotta Hallberg	Linda Wänström
f7kko	Kognitionsvet	Jalal Maleki	Mattias Arvola
6cddd	Datateknik		DMnämnden

# Konvertering från ER-diagram till relationer

# Studentförening: ER-diagram



# Från ER-diagram till Relationsmodell

1. Entitetstyper
2. Sambandstyper
3. Markera nycklar i relationsschemat
4. Specifiera Integritetsvillkor
5. Kontrollera redundans

# 1. Entitetstyper

Entitetstyp => relation

(varje entitetsinstans blir en tupel i relationen!)

- Enkla attribut blir attribut (kolumner) (1)
- Sammansatta attribut representeras av delarna (8)
- Flervärda attribut blir en egen relation med nyckelattributet och det flervärda attributet enkelt (9)
- Svag entitetstyp får som extra attribut nyckeln till den ägande entitetstypen (7)

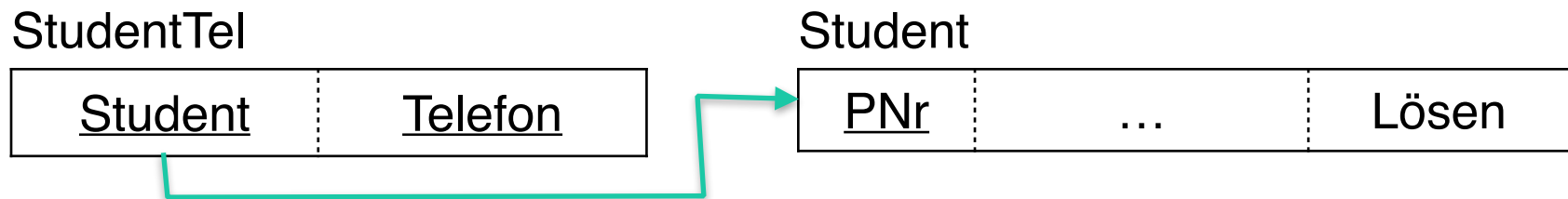
## 2. Sambandstyper

- Binära samband:
  - 1:N samband; lägg in 1-entitetens nyckel i relationen för N-entiteten (2)
  - 1:1 samband; lägg in nyckeln till den ena entiteten i den andras relation (3)
  - N:M samband blir en **egen relation** bestående av nycklarna till båda entiteterna (4)
- Flervägssamband blir en **egen relation** bestående av nycklarna till de sammanbundna entiteterna (5)
- Ägandesamband ignoreras!
- Attribut på samband ”följer med” (6)

## 2. Sambandstyper - referensattribut

Entitetstypers nycklar som läggs i andra relationer än den som representerar den egna entitetstypen blir referensattribut!

- t.ex. Student i StudentTel refererar till PNr i Student



### 3. Markera nycklar i relationsschemat

- ER-modellens nyckelattribut?
- Andra relationer – vad har de för nycklar?
- Kan det finnas alternativ?

# Nycklar: Definitioner

- Om något eller några av attributen i en relation kan användas för att identifiera hela tupeln så är det/de attributen en **nyckel**
- Typer: Supernyckel, kandidatnyckel, primärnyckel

# Supernyckel

- Om  $k$  är en delmängd av attributen i en relation  $R$
- sådan att  $k$  kan användas för att identifiera tuplerna i  $R$
- så är  $k$  en **supernyckel** för  $R$ .
  
- Notera:
  - $k$  kan innehålla ”onödiga” attribut
  - en relation har normalt flera supernycklar

# Kandidatnyckel = minimal supernyckel

- En supernyckel  $k$  är **minimal** om vi inte kan ta bort något attribut ur  $k$  så att den nya  $k$ ,  $k'$ , fortfarande är en supernyckel.
- Generellt finns det flera kandidatnycklar för en relation.

# Primärnyckel

- Den kandidatnyckel som **väljs** av databasdesignern som huvudnyckel för en relationstabell R kallas **primärnyckel** eller **nyckeln** till R.
- Används i andra relationer för att referera till en viss tupel i R (referensattribut).
- Primärnyckeln markeras i relationsschemat med understrykning

### 3. Markera Nycklar i relationerna - regler?

Relationer baserade på:

- vanlig entitetstyp
- svag entitetstyp
- flervärt attribut

Får som primärnyckel:

- nyckeln ur ER-diagrammet.
- nyckeln till den ägande entitetstypen plus den partiella nyckeln.
- nyckeln till entitetstypen **plus** attributet självt.

### 3. Markera Nycklar i relationerna - forts.

Relationer baserade på:

- binär sambandstyp,  
M:N
- flervägssambandstyp

Får som primärnyckel:

- nycklarna ur de två entitetstyperna
- nycklarna till de sammanbundna entitetstyper som kan ingå i flera sambandsinstanser

## 4. Specificera integritetsvillkor

- Rita ut främmande nycklar (referensintegritet)
- Totalt deltagande ger integritetsvillkor
- Kardinalitet för flervägssamband skrivs i text
- Domän för attribut (om inte tydligt från namnet).
- Semantisk integritet som fanns i ER-diagrammet (ex en student får ej vara registrerad på flera program samma år)

## 5. Kontrollera redundans

- Förenkla vid behov - finns samma sak på flera ställen?
- Normalisera (jfr föreläsning om detta)

# Studentförening: Relationer

Student (PNr, Namn, Gata, HusNr, PostNr, PostOrt, liu-id, Lösen)

StudentTel (Student, Tel)

Program (Kod, Namn, Programansv, Studierektor, Sektion)

Sektion (Namn, Lokal)

Fest(Sektion, Namn, tidpunkt, budget)

RegistreradPå (Student, Program, År)

MedlemI (Student, Sektion, BetTermin)

# Kokbok, modifierad

1. Entitetstyper -> relationer (kom ihåg alla attribut)
2. Sambandstyper (in i relationer eller blir egna relationer)
3. Markera nycklar
4. Integritetsvillkor
5. Redundanskontroll, normalisering

Frågor?

[eva.ragnemalm@liu.se](mailto:eva.ragnemalm@liu.se)

[www.liu.se](http://www.liu.se)