

# 732G16: Databaser, Design och programmering

Fö 1 Introduktion till kursen och ämnet

Eva L. Ragnemalm, IDA  
([eva.ragnemalm@liu.se](mailto:eva.ragnemalm@liu.se))

# Innehåll

- Kursöversikt
- Bakgrund
- Grundläggande begrepp

# Innehåll

- Kursöversikt
  - Innehåll
  - Kursinformation
  - Examination
  - Utvärdering
- Bakgrund
- Grundläggande begrepp

# Kursöversikt

- Innehåll: Teori om databaser, praktik
  - Litteratur: Kursbok, lektionshäfte, online-tutorial
  - Föreläsningar
  - Lektioner, labb, litet projekt

# Miniprojektet fokuserar på utredning

- Analysera data om studenter
  - Finns det samband mellan meritpoäng och hur många poäng man tar vid universitetet?
  - Skiljer det sig för obligatoriska kurser och valbara?
  - Skiljer det sig för olika program?
- Designa databas, implementera den (SQL), lägga in data (R), manipulera data (SQL), analysera data (R).

# Kursöversikt

- Innehåll
- Kursinformation
  - Kurshemsida <http://www.ida.liu.se/~732G16/>
  - Epost (ange kurskoden)
  - Lisam, Webreg
- Examination
- Utvärdering o ändringar

# Kursöversikt

- Teori och praktik
- Kursinformation
- Examination (Tenta + Labbkurs)
  - Tentamen: 4hp, ger kursens betyg. Salstenta.  
Datum kan ev. ändras!
  - Labbkurs 3,5hp (=100h)
  - datum för inlämningar/demo se TimeEdit.
- Utvärdering o ändringar

# Kursöversikt

- Teori och praktik
- Kursinformation
- Examination
- Utvärdering o ändringar



# Utvärderingen förra året

- 9 av 26 svarade ... Vad tyckte de andra ?
- Högt helhetsbetyg (4.67)
- Ett lektionstillfälle till (2 pers)
- ”Tydligare vad som ska göras i vilket moment i projektet, exempel på hur det ska se ut.”
- ”Upplägget på tentan”

# Innehåll

- Kursöversikt
- Bakgrund
- Grundläggande begrepp

# Bakgrund

# Varför databaser?

- Varför ni?
- Varför överhuvudtaget?
  - Spara utrymme (inte dubbellagra info, men ändå ha allt tillgängligt med rimlig åtkomsttid)
  - Förenklar lagring och hantering av data

## Exempel: telefonbok i R

```
>phones<- c("073-34598764", "013-223345",  
"07012334545", "011-4345456")  
  
>names<- c("Lasse", "Bosse", "Lisa", "Anna")  
  
>contacts <-data.frame(name = names, telnr =  
phones)  
  
>find_phone<- function(n, df) {  
  target<-0  
  for (x in (1:nrow(df)))  
    if (df[x, 1]==n){target<-x }  
  if (target>0) {print (df[target, 2])  
  } else print ("Not found")
```

---

# Telefonbok med databas

- Med databasspråket SQL:

Skapa databasen

```
create table contacts (  
    name varchar(20), telnr varchar(11));
```

- Lägga in data:

```
insert into contacts values ('Britta',  
    '070-123456')
```

- Söka Annas telefonnummer:

```
select telnr from contacts  
    where namn='Anna';
```

# Telefonbok, forts

- SQL:  
För att lägga till födelsedatum:  
**alter table contacts add  
(bdate date);**
- För snabbare sökning: index  
**create index nrIndex on  
contacts(telnr);**
- Logiskt och fysiskt dataoberoende

# Innehåll

- Kursöversikt
- Bakgrund
- Grundläggande begrepp



# 732G16 Databaser, design och programmering

Databasteori:  
definitioner

# Databas

En **databas** är en samling **data** som:

- Har med varandra att göra
- Representerar (modellerar) en del av verkligheten för ett specifikt ändamål
- Är beständig (finns kvar)
- Har en strikt struktur som beskrivs i ett **schema**
- Är ”**korrekt**”
- Lagras på dator

# Databashanteringssystem (DBMS)

Ett programsystem som

- **Hanterar** flera databaser
- enligt en **datamodell**
- med hjälp av ett **databasspråk**.
- Kan anropas från **applikationsprogram**

# Hantera databaser

- Skapa nya databaser
- Manipulera data (lägga till, ta bort, läsa ut)
- Kontrollera korrekthet (**integritet**)
- Säkerhet

# Hantera databasen: **Databasintegritet**

- Informationen i en databas måste vara korrekt!
  - Rätt typ och rätt struktur, ett av en mängd
  - Värden som måste fyllas i
  - Överensstämmelse med andra värden
- Korrekthetsvillkor i db: integritetsvillkor
- Formuleras mha olika mekanismer i DBHS

# Hantera databasen: Säkerhet

- Återhämtning vid fel i hård/mjukvara
- Skydda data (inloggning, begränsningar)
- Hantera samtidig åtkomst mot flera användare



Vy

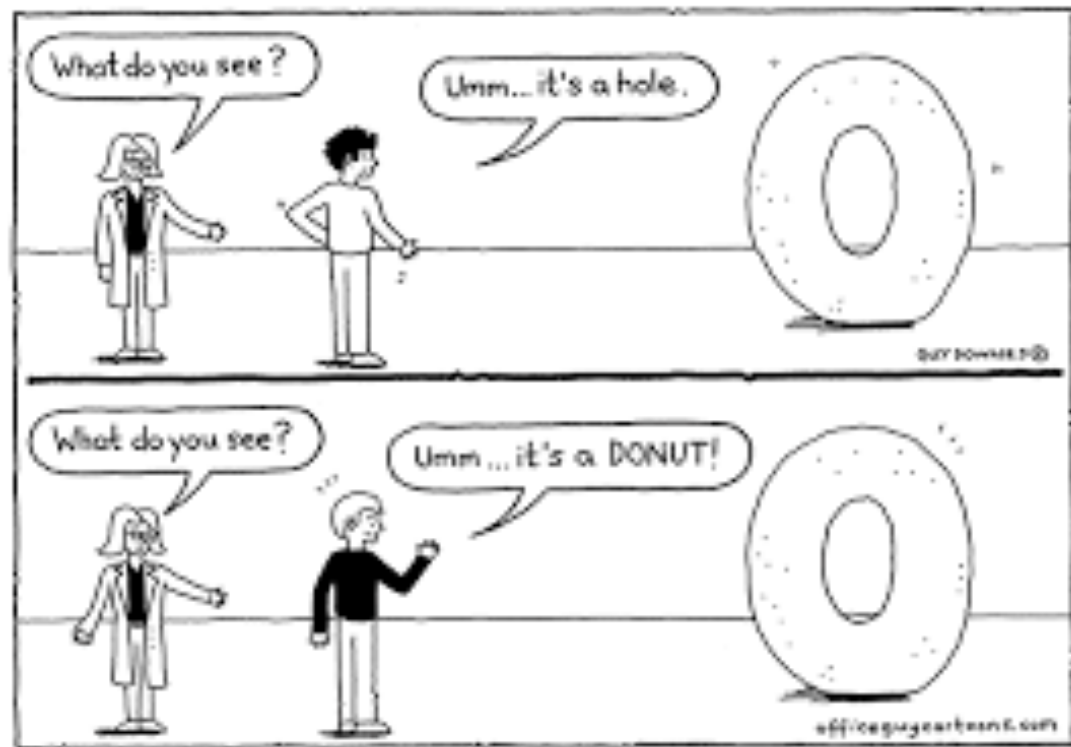
# Databashanteringssystem (DBMS)

Ett programsystem som

- **Hanterar** flera databaser
- enligt en **datamodell**
- med hjälp av ett **databasspråk**
- Kan anropas från **applikationsprogram**

# Datamodell

- Hur vi ser världen







# Datamodell

Finns på olika detalj/abstraktions nivåer:

- Konceptuell (begreppsmässig) modell
- Representationsmodell (implementationsmodell)
- Fysisk datamodell

# Datamodell: Representationsmodeller

- Hierarkisk modell
- Nätverksbaserad
- Relationsmodellen
- Objektorienterad
- Grafdata

# Datamodell: **Relationsmodellen**

## Student

Epost	Namn	Adress	Kurs
Annan123	Anna Andersson	Rydsv 234	729G68
Ollol345	Olle Olsson	Storgatan 1	732G16

## Kurs

Kurskod	Namn	Läsår	Poäng
729G68	Databaser	07/08	7.5
732G16	Databaser	07/08	7.5
TTIT03	Introduktion till PBL	07/08	2

# Datamodell: **Schema** och **Instans**

- Databasens beskrivning och dess innehåll
- Instans = den samling data som finns i databasen vid en viss tidpunkt.
- Schema=beskrivning av databasen enligt någon datamodell. Förändras sällan.
  
- Innehållet måste följa schemat.
- Schemat lagras i databasen

# Datamodell: **Metadata**

Data om data

Exempelvis:

- Schemat
- Information om hur och var data lagras
  - Struktur, organisation
- Tillgänglighet
- Användningsstatistik

# Datamodell: Metadata - exempel på användning

- Användare:
  - Vilken information finns i databasen
  - Tidsåtgång för att ta fram info Y
- Databasadministratören:
  - Hur mycket utnyttjas olika delar?
  - Vilka svarstider har vanliga frågor?
  - Behövs optimering?
  - Har någon försökt knäcka säkerheten?

# Databashanteringssystem (DBMS)

Ett programsystem som

- **Hanterar** flera databaser
- enligt en **datamodell**
- med hjälp av ett **databasspråk**.
- Kan anropas från **applikationsprogram**



# Databasspråk

Klass av språk som används mot databaser

- Databeskrivningsspråk (**DDL**)
- Datamanipuleringspråk (**DML**)
- Värdspråk

# Databasspråk: Exempel: **sql**

- **Create table** person (pnr **varchar**(10) **not null**, namn **varchar**(30), adress **varchar**(30), **primary key**(pnr))
- **Update** person  
set adress 'Studentvägen 10'  
where pnr='7012120123'
- **Select** namn **from** person  
where pnr='7012120123'

# Databashanteringssystem (DBMS)

Ett programsystem som

- **Hanterar** flera databaser
- enligt en **datamodell**
- med hjälp av ett **databasspråk**
- Kan anropas från **applikationsprogram**

# Applikationsprogram

- Specialiserade program som gör användningen av databasen enklare för slutanvändare
- Specifik användargrupp och specifikt användningsområde
- Ex online-butiker, Ladok, TimeEdit

# Databashanteringssystem (DBMS)

Ett programsystem som

- **Hanterar** flera databaser
- enligt en **datamodell**
- med hjälp av ett **databasspråk**.
- Kan anropas från **applikationsprogram**

# Fördelar med databasteknik

- Självbeskrivnade
- Dataoberoende
- Säkerhetssystem (integritet, tillgänglighet, loggning)
- Stöder och reglerar samtidig användning
- Motverkar redundans

# Nackdelar med databasteknik

- DBMS tar stor plats
- Overhead ger längre svarstider
- Ibland krångligt att lära sig hantera

# När ska man **inte** använda DBMS

- Små mängder data
- Enstaka användning av data som sedan blir föråldrat/förkastat
- Enkel användning som aldrig ändras
- Om systemet måste ta liten plats
- Om systemet måste garantera korta svarstider (processövervakning etc)



# Summering viktiga termer

- Databas
- databashanterare, dataoberoende
- databasintegritet
- datamodell (3 nivåer), ER-modell, relationsmodell
- Instans, schema, metadata
- Databasspråk, DDL, DML, SQL
- Applikationsprogram

Frågor?

Nästa föreläsning: ER-modellen

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