

732G16: Databaser, Design och programmering

Fö 1 Introduktion till kursen och ämnet

Eva L. Ragnemalm, IDA
(eva.ragnemalm@liu.se)

Innehåll

- Kursöversikt
- Bakgrund
- Grundläggande begrepp

732G16 Databaser, design och programmering, fö 1.

Kursöversikt

Kursöversikt

- Innehåll
- Kursinformation, kommunikation
- Examination
- Schema
- Kursutvärdering

Innehåll i kursen

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

Innehåll, forts

- Litet projekt, en utredning:
 - 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).

Kursinformation

- Kursinformation
 - Kurshemsida: www.ida.liu.se/~732G16
- Kommunikation:
 - Epost (ange kurskoden)
- Andra system:
 - Lisam, för inlämningar
 - Webreg, för resultat

Examination

- Examination: Tenta + Labbkurs
 - Tenta: Salstenta. 4hp, ger kursens betyg.
 - Labbkurs: 3,5hp (=100h). Datum för inlämningar/
demo se TimeEdit.

Schema

- Timeedit
- Ändringar? Omtentor, sektionsaktiviteter?

Kursutvärdering

- Evaluate:
 - 6 av 35 svarade ... Vad tyckte de andra?
 - Ovanligt lågt helhetsbetyg (2.8)
 - Kommentarer:
 - Stökigt på labbarna (?)
 - Assistent som inte rättade rätt
 - Pedagogiska genomförande stöttade ej

Innehåll

- Kursöversikt
- Bakgrund
- Grundläggande begrepp

732G16 Databaser, design och programmering, fö 1.

Bakgrund

Varför databaser?

- Varför ni?
- Varför överhuvudtaget?
 - Spara utrymme (inte dubbellagra data, 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(name, df) {  
  target<-0  
  for (x in (1:nrow(df)))  
    if (df[x, 1]==name){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

Mer fördelar med databasteknik

- Ingen dubbellagring av data
- Säkerhet
 - Åtkomst
 - Backup och återställning vid krascher
 - Samtidig användning
 - Korrekt data

Innehåll

- Kursöversikt
- Bakgrund
- Grundläggande begrepp

732G16 Databaser, design och programmering fö 1

Grundläggande begrepp inom Databaser

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**

- Data 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

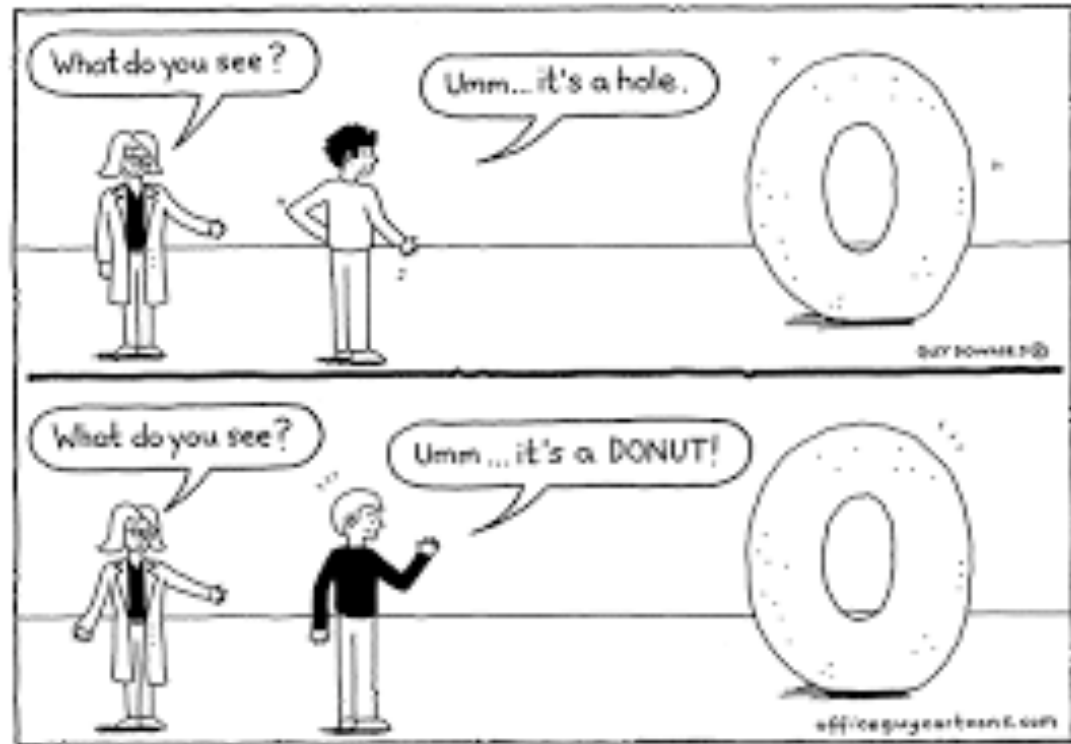
Databashanteringssystem (DBMS)

Ett programsystem som

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

Datamodell

- Hur vi ser världen



Datamodell

- Hur vi ser världen



Datamodell

Finns på olika detalj/abstraktions nivåer:

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

Datamodel: **Representationsmodeller**

- Hierarkisk modell
- Nätverksbaserad
- Relationsmodellen
- Objektorienterad
- Grafdata

Datamodell: **Relationsmodellen**

Student

Epost	Namn	Adress	Kurs
Annan123	Anna Andersson	Rydsv 234	729G28
Ollol345	Olle Olsson	Storgatan 1	732G16

Kurs

Kurskod	Namn	Läsår	Poäng
729G28	Databaser	25/26	6
732G16	Databaser	23/24	7.5
TDDD60	Interaktiva system	24/25	4

Datamodell: **Schema**

- Beskrivning av databasens struktur (enligt någon datamodell)
 - För en relationsmodell: tabeller, kolumner, värde
- Schema vs Instans - den samling data som finns i databasen vid en viss tidpunkt.
- Instansen (innehållet) måste följa schemat.
- Schemat lagras i databasen

Datamodell: **Metadata**

Data om data

Exempel:

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

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
- Exempel: 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

- Självbeskrivande
- 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
- Schema, instans, metadata
- Databasspråk, DDL, DML, SQL
- Applikationsprogram

Frågor?

Nästa föreläsning: ER-modellen

www.liu.se