

# Intro till Databaser

## Vad är en databas?

*"En persistent samling data som hör ihop och modellerar en del av världen"*

*-ThomasPadron-McCarthy*

<https://www.youtube.com/watch?v=wR0jg0eQsZA>

# Databaser

En databas ska ha en beskrivning över informationen som den lagrar

**Sk.Schema**

Informationen (datan) får inte heller vara motsägelsefull. Den ska vara **konsistent**

*Exempel: samma person finns två gånger i samma databas fast med olika hemadresser. Vilken är rätt?*

# Databashanterare

Om en databas är en samling data så behöver vi nåt sätt att hantera datan på.

Vi behöver en **databashanterare** (eng. **databasemanager**)

En databashanterare är ett program (eller oftast ett system av flera program) som har till uppgift att lagra och hantera databaser.

Det finns en uppsjö olika databashanterare där ute: MySQL, SQLite, PostgreSQL, MS Access, MS SQL Server mm. mm.mm....

# Varför Databaser?

Hittills har vi bl.a lagrat data i vanliga hederliga textfiler, vad är det för fel med det?

- Inget fel egentligen, men dedikerade databaser är oftast **enklare**, **kraftfullare** och mer **flexibelt** än att läsa/ skriva från textfiler.

# Enkelt

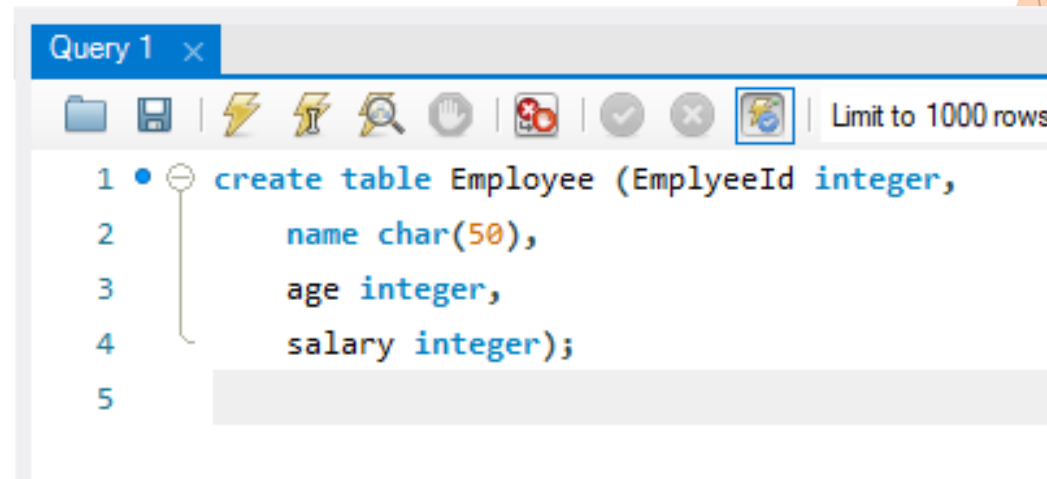
Att lägga till data till en databas är relativt enkelt

Säg att vi har objekt av klassen employee som vi vill lagra i en databas

Bilden nedan skapar en tabell av employee objekt i en databas som vi sedan kan manipulera

(Skrivet med SQL i MySQL)

```
public class Employee {  
    int EmployeeId;  
    String name;  
    int age;  
    int salary;  
}
```



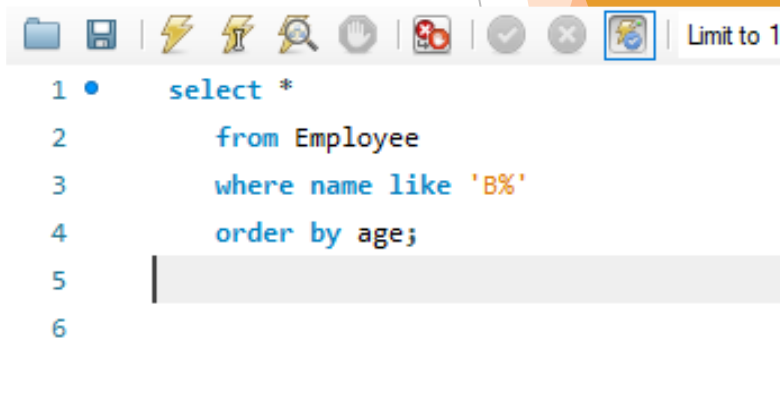
# Kraftfullt

En databas har en uppgift: att lagra och hantera data.

Det betyder att den ägnar **alla** sina resurser till detta ändamål och det blir därför **snabbare** och inte minst **stabilare** än att göra detta i exempelvis ett vanligt javaprogram som skriver till textfiler.

Det finns inbyggda kommandon och algoritmer för att lista, sortera, lägga till, radera etc. i databashanterare

Till höger så vill vi hitta alla Employee som börjar på B och sortera dessa efter ålder.

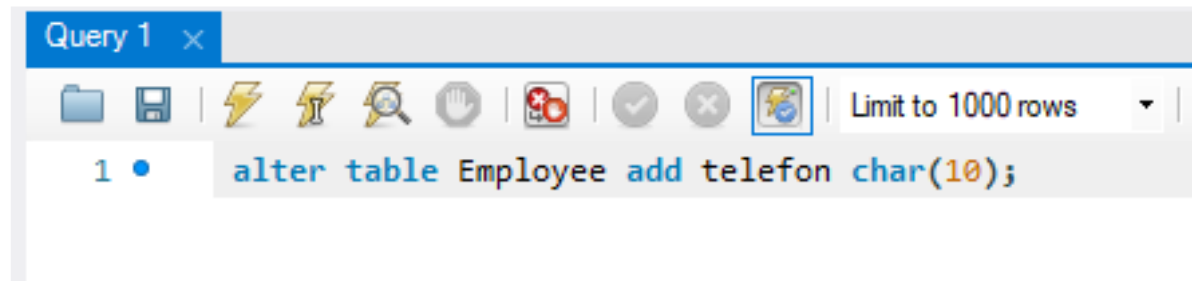


```
1 • select *
2     from Employee
3     where name like 'B%'
4     order by age;
5
6
```

The screenshot shows a SQL query editor window with a toolbar at the top containing icons for file operations, execution, and search. The query text is as follows:

# Flexibelt

Eftersom man (oftast) jobbar med tabeller så är det enkelt att lägga till och ta bort kolumner. Säg att vi vill lägga till telefonnummer till våra employees. Hade vi gjort det med java och textfiler så hade det krävt en del meck att uppdatera filen. I en databas kan man oftast göra det men en enkel kodrad.

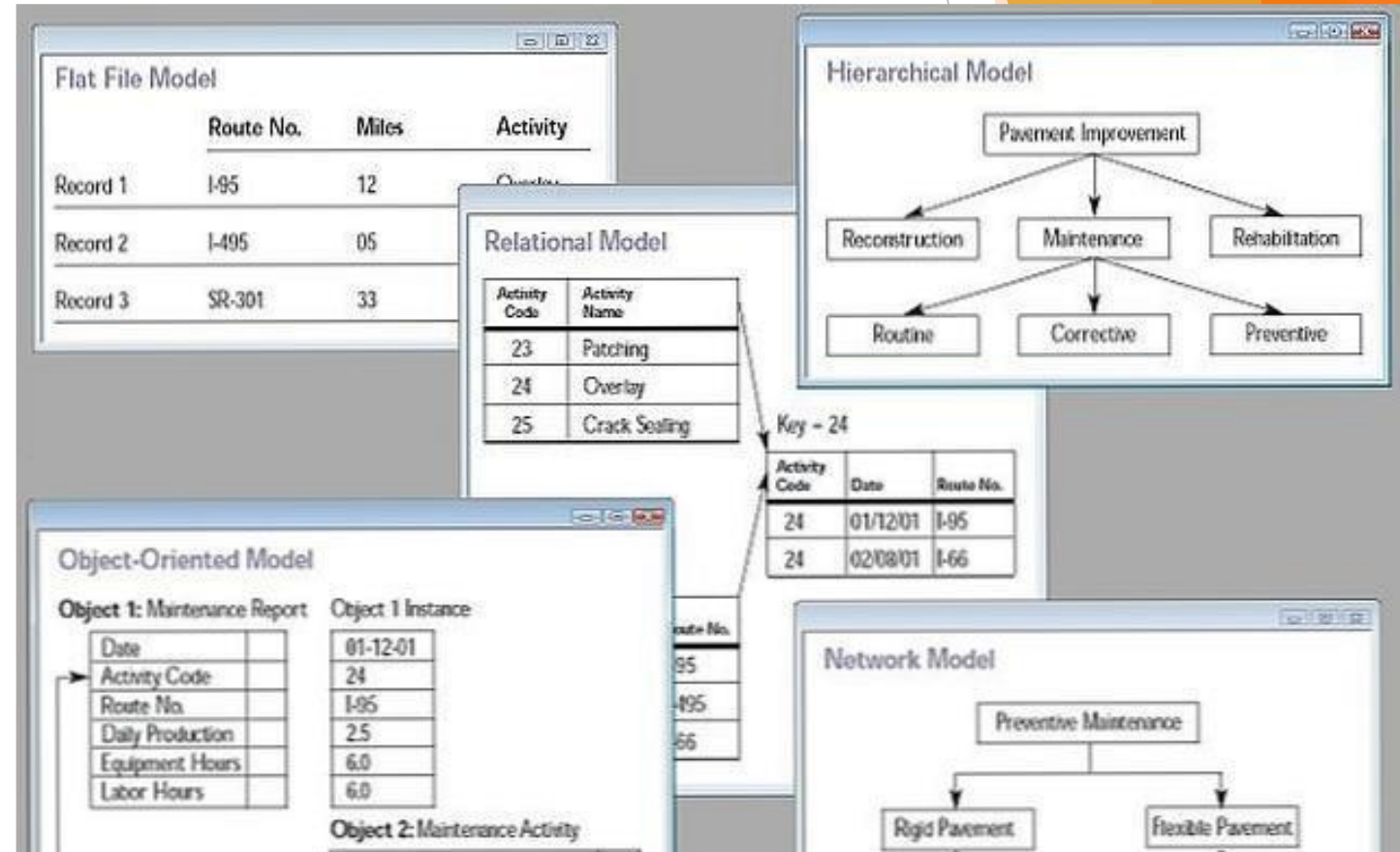


# Datamodell

## Vad är en *datamodell*?

En *datamodell* är ett sätt att organisera data, i exempelvis en databas.

Det finns olika typer av datamodeller, precis som det finns olika typer av databaser





# Relationsmodellen

Relationsmodellen är den vanligaste datamodellen för databaser och den vi kommer titta närmare på nu.

Modellen går ut på att man lagrar data i tabeller (kallas **relationer** eller **entiteter**) som refererar till varandra.

Raderna brukar kallas **tupler**.

Kolumnerna brukar kallas **attribut**.

Det får *inte* finnas två identiska tupler i samma entitet/relation

## Exempel på en relation

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

attribut  
(eller kolumner)

tupler  
(eller rader)

relationens  
namn

*instructor*

# Relationsmodellen

I exemplet till höger så representerar varje rad en lärare.

Varje kolumn anger ett attribut som läraren har.

Den markerade läraren har med andra ord attributen ID = 15151, name= Mozart, dept\_name= Music, salary=40000

## Exempel på en relation

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

attribut  
(eller kolumner)

tupler  
(eller rader)

relationens  
namn

*instructor*

# Nycklar

I relationer anger man även *nycklar*.

Nycklar är attribut vars värden är unika. Det kan antingen vara ett enskilt attribut, eller en kombination av attribut.

I exemplet till höger kan exempelvis ID vara en nyckel, eftersom det inte finns 2 likadana ID.

En nyckel skulle även kunna vara ID+name, eftersom det fortfarande inte finns 2 rader med samma värden på ID+name

## Exempel på en relation

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

attribut  
(eller kolumner)

tupler  
(eller rader)

relationens  
namn

*instructor*

# Nycklar

De viktigaste nycklarna att hålla reda på är *primärnycklar* (*primary key*) och *främmande nycklar* (*foreign key*).

Man brukar även prata om *supernycklar* och *kandidatnycklar*, men det är lite överkurs just nu.

## Exempel på en relation

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

attribut  
(eller kolumner)

tupler  
(eller rader)

relationens  
namn

*instructor*

# Nycklar

Varje relation ska ha en **primärnyckel**, och det är oftast den vi använder för att identifiera tupler i relationen

En primärnyckel är oftast en integer men det är möjligt att använda andra datatyper (ex. text)

I exemplet till höger är ID primärnyckel.

## Exempel på en relation

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

attribut  
(eller kolumner)

tupler  
(eller rader)

relationens  
namn

*instructor*

# Kopplingar mellan relationer

Oftast så har man flera tabeller (relationer) i en databas, och dessa vill man koppla till varandra.

Antag att vi har en sportklubb med 3 olika relationer: Medlemmar, Sektioner och Deltagare

Medlemsnummer	Namn	Telefonnummer
2	Stina	282677
3	Saddam	260088
4	Lotta	174590
1	Olle	260088

Sektionskod	Namn	Ledare
A	Bowling	4
C	Konstsim	2
B	Kickboxing	4

Medlem	Sektion
1	A
1	B
1	C
2	C
3	A

# Kopplingar mellan relationer

I första relationen anger vi medlemsnummer som primärnyckel.

I andra relationen anger vi sektionskod som primärnyckel.

Ledare i andra relationen hänvisar till medlemsnummer i den första relationen. Ledare är en sk **referensattribut** eller **foreign key** då den hänvisar till en primärnyckel i en annan relation.

På detta sätt kan vi mao se att telefonnumret till ledaren för sektion A är 174590

Medlemsnummer	Namn	Telefonnummer
2	Stina	282677
3	Saddam	260088
4	Lotta	174590
1	Olle	260088

Sektionskod	Namn	Ledare
A	Bowling	4
C	Konstsim	2
B	Kickboxing	4

# Kopplingar mellan relationer

I den tredje relationen kan vi se vilka medlemmar som deltar i de olika sektionerna.

Medlem är referensattribut till första relationen och sektion är referensattribut till andra relationen.

Vad är primärnyckeln i tredje relationen?

Medlemsnummer	Namn	Telefonnummer
2	Stina	282677
3	Saddam	260088
4	Lotta	174590
1	Olle	260088

Sektionskod	Namn	Ledare
A	Bowling	4
C	Konstsim	2
B	Kickboxing	4

Medlem	Sektion
1	A
1	B
1	C
2	C
3	A



# Länkar

Det vi gått igenom här kommer från en bok om databasteknik. Delar av den finns även i slimmad version på digital form:

<http://databasteknik.se/webbkursen/index.html>

<http://databasteknik.se/webbkursen/databaser/index.html>

<http://databasteknik.se/webbkursen/relationer/index.html>