



Modul DB1-3

Datamodellering

Antal föreläsningar: **2**

Antal laborationer: **1**

Förkunskapskrav: **Grundläggande kännedom
om databaser (Modul DB1-2)**

Kurslitteratur: **“Praktisk datamodellering”**
ISBN: 91-44-38001-1

Referenslitteratur:

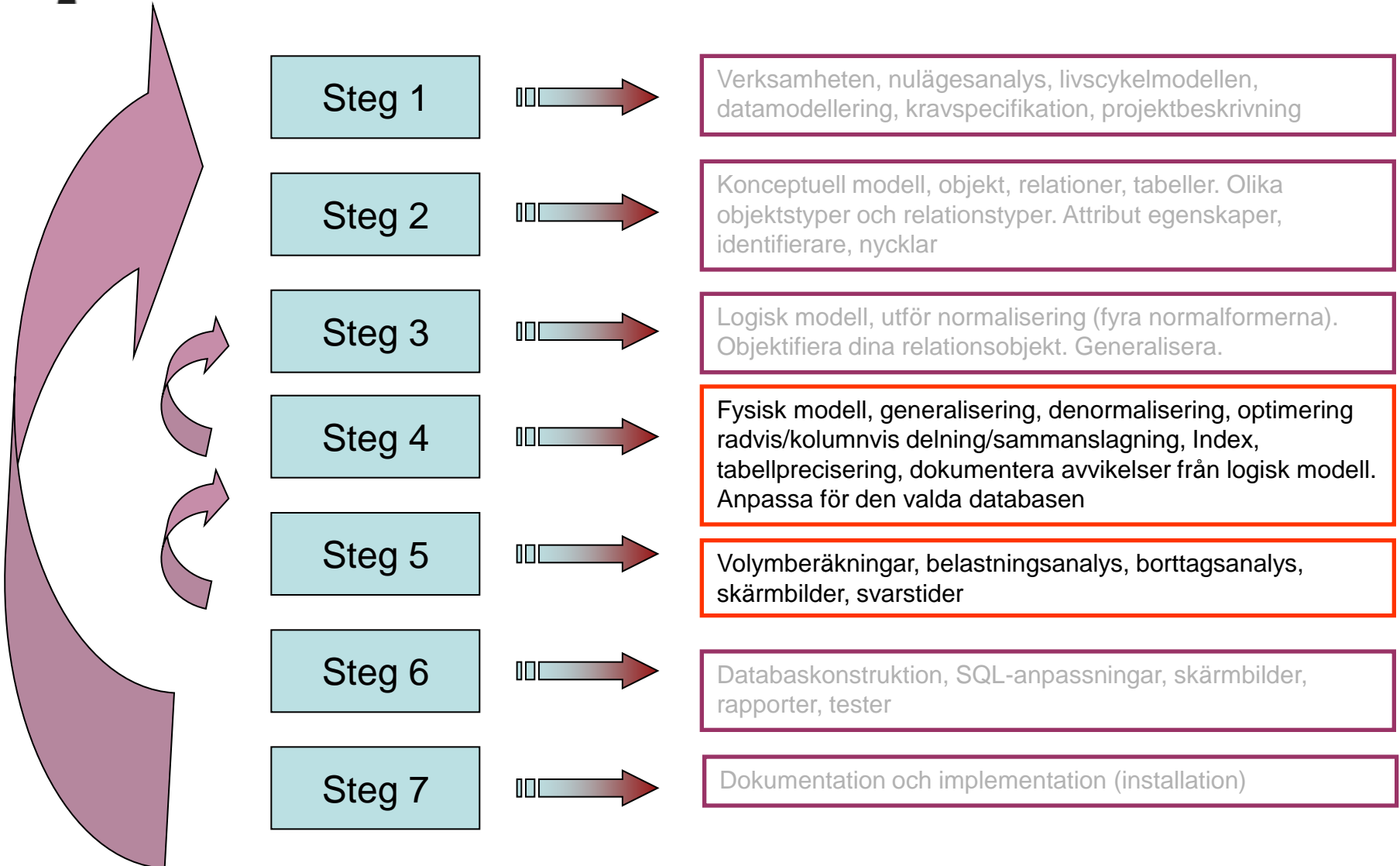


Innehållsförteckning:

- Optimering / Denormalisering
- Kolumnvis delning
- Radvis delning
- Kolumnvisa sammanslagning
- Redundanta tabeller
- Index
- Referentiell Integritet
- Volymberäkning
- Tillväxt
- Historik



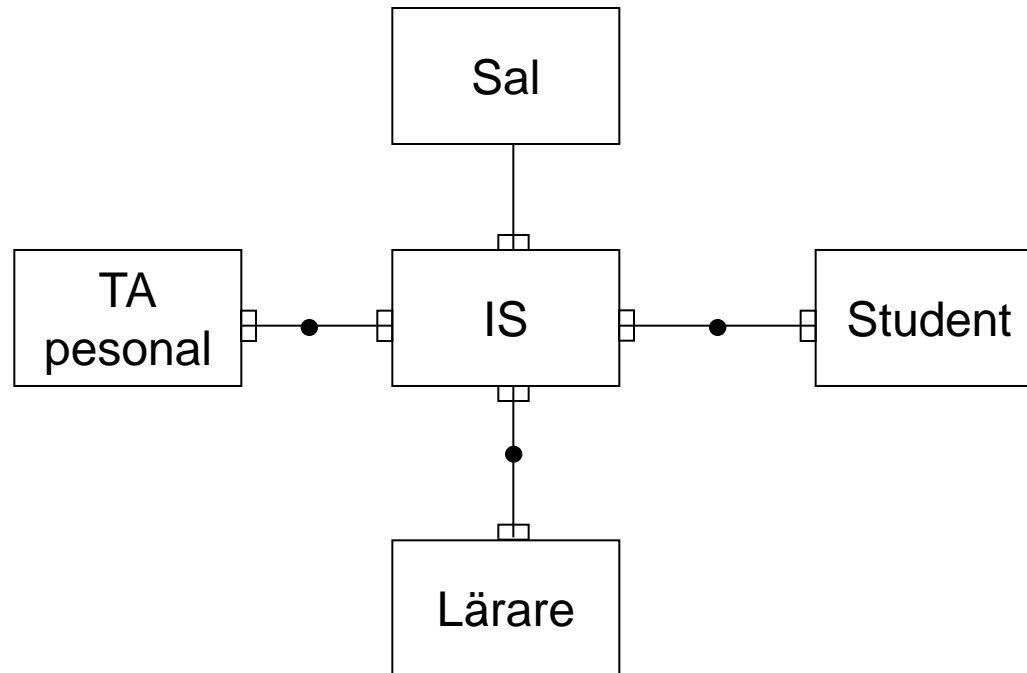
Utvecklingsprocessen





Generalisering

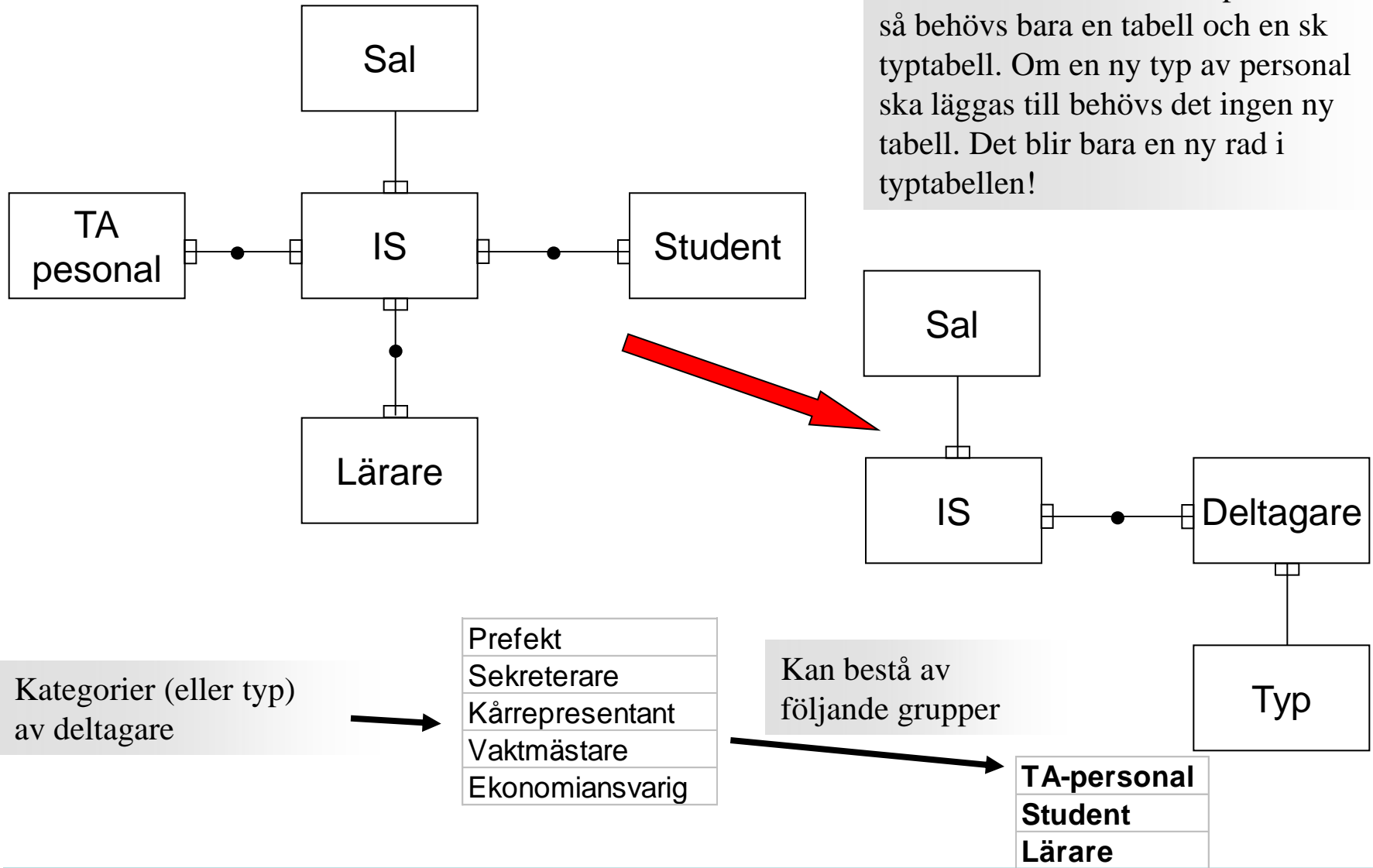
I samband med modellering av verksamheten för IS (InstitutionsStyrelsen) inom teknikinstitutionen fick fram följande modell.



Inom varje institution finns en institutionsstyrelse med medlemmar från olika kategorier av personal och studenter.



Generalisering



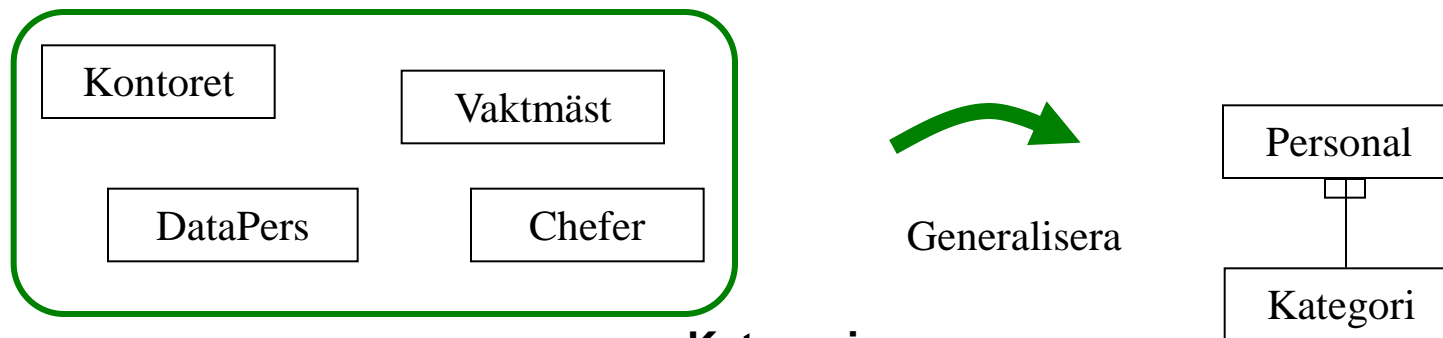


Generalisering

- Mål:** • **Systemet ska kunna växa** • **Förenkling** • **Öka prestanda**
Problem: *Användarna känner inte igen sig (Vänta till efter seminariet)*

Fall inte för kundens beskrivning av verksamheten och skapa objekt för *kontors-personal*, *vaktmästare*, *chefer*, och *datafolk*.

Vanligen kan *alla kategorier inordnas under* begreppet *personal* med en kompletterade kategori-tabell.



Personal

PersID	Namn	PkatID	
10	Percy	3	
11	Erik	2	
12	Sten Å	2	

Kategori

PkatID	Benämning
1	Chef
2	Kontor
3	Data



Optimering Kolumnvis delning



Då viss information sällan efterfrågas kan man dela tabellen i två tabeller
"Viktiga data" + "Sällan använda data" => Mindre och snabbare tabeller

Kund

KundID	Namn	Bla Bla	Noteringar
14	AB S&P	...	Diverse för det mesta ointressant information
20	Context AB	...	

KundA

KundID	Namn
14	AB S&P
20	Context AB

KundB

KundID	Bla Bla	Noteringar
14	...	Diverse för det mesta ointressant information

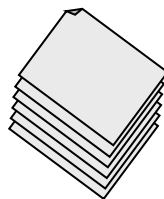
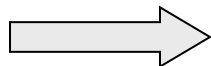
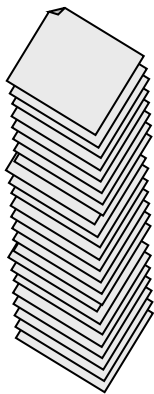
Kan också vara ett sätt att separera publik data från säkerhetskänslig data.
=> **Säkrare tabeller**

**Kolumnvis delning
(Dekomponering)**

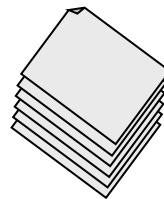


Optimering Radvis delning

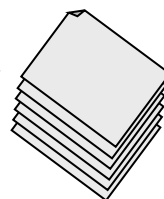
0,4milj
rader



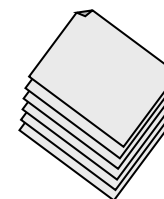
0,1 milj
rader



0,1 milj
rader



0,1 milj
rader



0,1 milj
rader

Tabellerna delas upp i flera kortare tabeller

- Aktuell info / Gammal info (ex årets fakturor o tidigare års fakturor)
- En tabell per år
- Aktiva / Passiva kunder

Fördelar:

- Korta tabeller - Snabb access
- Lätt att organisera arkivering av gamla data

Nackdelar:

- Måste byta tabeller
- Svårare att göra statistik över flera tabeller.

Radvis delning
(Segmentering/Partitionering)



Optimering Kolumnvis sammanslagning

Slå samman två eller flera normaliserade tabeller till en onormaliserad.

Kund

Kundnr	Namn	Postadress	Tel	Distrikt
1	Direct AB	Gatan 33 Hjo	123 44	01
2	Direct LTD	Vägen 33 Ystad	55 66 77	03

Distrikt

Distrikt	Distriktsnamn
01	Mellan
03	Södra



Kund

Kundnr	Namn	Postadress	Tel	Distrikt	Distriktsnamn
1	Direct AB	Gatan 33 Hjo	123 44	01	Mellan
2	Direct LTD	Vägen 33 Ystad	55 66 77	03	Södra

Fördelar:

- Snabbare access
- Enklare – minskar antalet joinoperationer

Nackdelar:

- Ökad redundans
- Problem vid uppdatering av distriktnamn
- Svårighet att lagra distrikt utan kunder
- Applikationen måste göra mer omfattande validering

Man kan också dubbel-lagra distrikts-tabellen för att underlätta uppslag.



Optimering Reduntanta tabeller

Variant på denormalisering

- frekventa omfattande frågor lagras fysiskt i en tabell som kan läsas snabbt .
- måste genereras om då bakomliggande tabeller ändras.

Student - Alla studenter

StudID	Namn	Postadress	Tel	ProgID
1	Anders			1
2	Stina			2
7	Anna			1

Program - Utbildningslinje

ProgID	Program
1	Maskiningenjör
2	Webbprogrammerare

Maskin

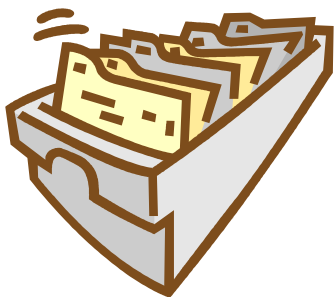
StudID	Namn	Program
1	Anders	Maskiningenjör
7	Anna	Maskiningenjör

Redundant tabell med studenter i viss klass.



Index, vad är

Index är egentligen ett annat namn på sorteringsordning.



Ett kartotek har en viss sorteringsordning för att vi lätt ska hitta våra data.



Data i en pärm ligger inordnade i flikar för att vi lätt ska hitta data som vi söker.

Sköter vi inte inläggningen av data på ett riktigt sätt så funkar det inte längre. Vi hittar inte det vi söker. Tappar vi kartoteket så det går indexet sönder.....



Ett index kan liknas vid ett sakregister i en bok.

En telefonkatalog är indexerad
på efternamn, förnamn

eller på telefonnummer

Gårdby Bil och Smide S. Sandby 2424 FÄRJESTADEN	333 33
Gårdby Frisering Kummelv. 14 FÄRJESTADEN	331 56
GårdbyBuss Gårdby FÄRJESTADEN	330 70
Gällby Bygg Gällby LÖTTORP	235 22
Gärdslösa församling/församlingshemmet	
Gärdslösa kyrka	56 02 39
Gärdslösa Hembygdsförening Jämjö	56 00 58
Gärdslösa-Långlöt-Runstens församlingar	
Se Svenska kyrkan	
Gästbåthamn, Strand Marina Gästhamnen	779 94
Görans TV-Service Borgv. 6	775 36
Götes Maskinstation Box 96, 380 74 LÖTTORP	203 88
H-L Försäljning Storg. 24	129 60
H & M Olands Köpstad FÄRJESTADEN	392 01, 392 00
H P Fastighetsservice Tallv. 3 LÖTTORP	208 77
H S B Bostadsrättsföreningen Kronan Torsg.	136 47
H S B Bostadsrättsförening Guntorp o. Tomaten	
Guntorpsg. Gårdshus	775 11
H S B Bostadsrättsfören. Solrosen, Stranden o.	
Kvarnen Kvarng. 25 vicevärd exp.	107 71
H T F Handelstjänstemannaförbundet	
Honnörsg. 26 Box 447, 351 06 VÄXJÖ Växel	0470-70 41 90
Haga Park Camping St. Frö	
MÖRBYLÅNGA Reception o. camp.	360 30
Hagaborg, Pensionat Pensionatsv. 1 KÖPINGSVIK	720 46
Half Price Storn. 24	774 50

95 Ponjonen Birgitta
98 Melkersson Öste

121

01 I F K Borgholm
02 Eriksson Elsa
03 Davidsson Jeanet
04 Nilsson Gustav
07 Eneman Ylva
08 Larsen Inger-Mar
09 Hammarsten Folk
10 Lillegren Håkan
11 Elmquist Olof
14 Larsson Lars-Olof
15 Sundberg Isabell
16 Bladh Lennart
17 Sjöberg Leif
18 Molander Stefan
19 Heigeresson Jörgen
20 Johansson Lars
20 Simonsson Yvonn
21 Johansson Staffa
24 Nilsson Birger
27 Borgholms Kem-

74 Lindfors Göran
75 Carlsson Gudrun
76 Björklund K A
77 Restaurang Vård
79 Andersson Tomm
81 Nilsson Gerd
82 Johansson Sonny
84 Thoreb AB
85 Hultenius Birger
86 Sjöberg Siverth
88 Nilsson Magnus
89 Carlsson Osten
91 Carlsson Evelyn
93 Mittbrodt Karl-Jo
94 Mittbrodts Möbel
95 Månsson Evy
96 Verneresson Håka
97 Erlandsson Christ
98 Johansson Susan
99 Tornegård Anna-

123

14 Hansson Maria
15 Kommunens Bäst



Index, olika typer

Oindexerat (heap) Långsam metod

Kundid	Namn
1	Kalle
2	Otto
3	Anna
4	Stina
5	Sture

Telid	Kundid	Tel
1	1	123123
2	2	456456
3	5	234341
4	3	124546
5	1	234560
6	4	988654
7	1	234987

Sökning sker sekventiellt (från 1:a post till sista post) genom hela telefon-tabellen för att hitta alla telefonnr som relaterar till sitt kundid.



Index, olika typer

Indexerat (sorted/hash) Snabb metod

Kundid	Namn
1	Kalle
2	Otto
3	Anna
4	Stina
5	Sture

Telid	Kundid	Tel
1	1	123123
5	1	234560
7	1	234987
2	2	456456
4	3	124546
6	4	988654
3	5	234341

Sökning sker med index och hittar alla telefonnr som relaterar till Kundid snabbt.



Struktur hos ISAM-filer (Index-Sekventiell Access Metod)

Index på Efternamn

Sorterad på efternamn
varje rad har en pekare
till motsvarande post i
huvudtabellen

Tabell Personal

Här ligger posterna i fysisk ordning
(I den ordning de skrevs in)

Enamn	Rec.nr
Albertsson	5
Askberg	7
Bredrup	3
Dyvert	1
Fransson	4
...	
Åkerstad	2

Rec.nr	Enamn	Fnamn	Adr	Riktnr	Tel	Ort
1	Dyvert	Bo	Gat..	480	233221	Kalmar
2	Åkerstad	Per	Väg..	470	333333	Växjö
3	Bredrup	Eva	Stig..	485	443122	Borgh..
4	Fransson	Siw	Box..	480	134223	Kalmar
5	Albertsson	Rut	Gat..	480	232323	Kalmar
6	...					
7	Askberg	Åke	Väg..	470	232323	Växjö

Det går snabbt att söka efter t ex Askberg i den sorterade indexfilen och hitta resten av posten via pekaren.



Hur kan ett index fungera

Tänk på ett tal mellan 1-10

Hur hittar man lättast det tänkta talet.....

Frågar man: är det 1?

 är det 2?

 är det 3?

Man kan ju ha tur..... Men

Eller ska man fråga: Är det större eller mindre än 5?

Nej - > då är det fem

Mindre än fem - > 1-4

Större än fem - > 6-10

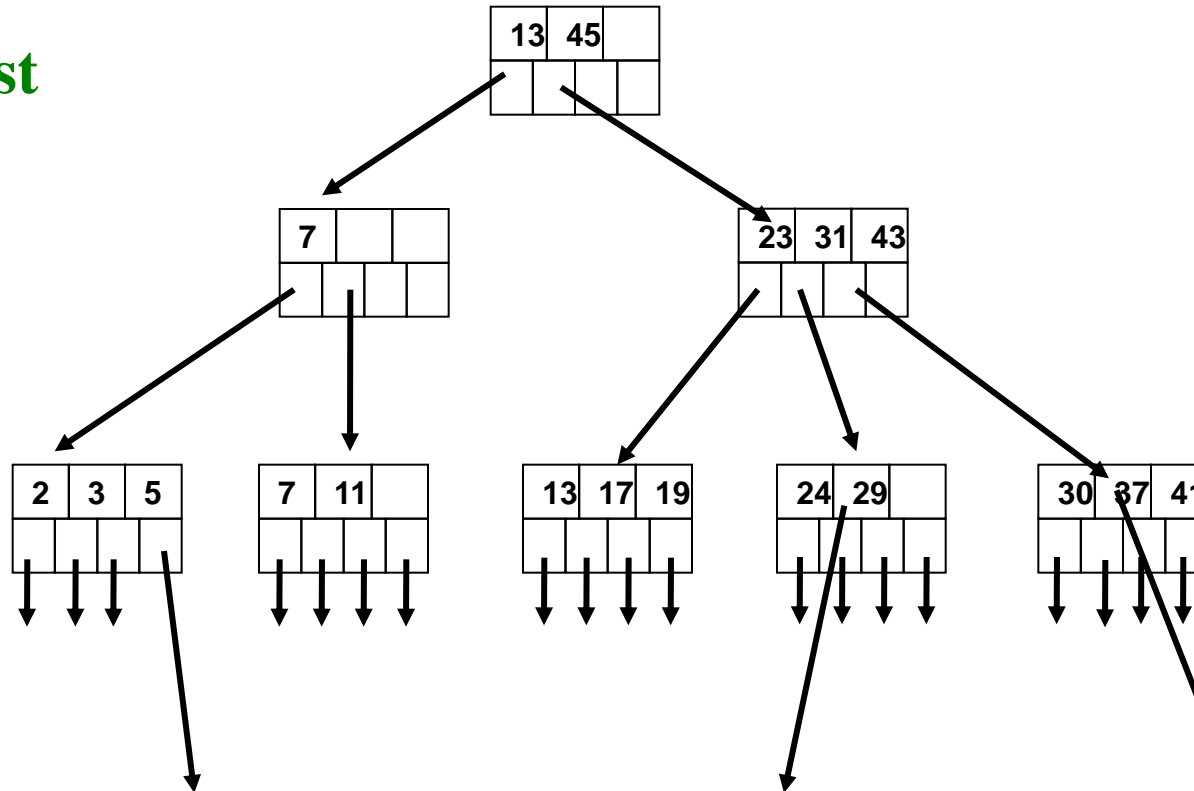


Index Balanserade träd

Index byggs vanligen med balanserade träd, B*-träd

Direktåtkomst
via index

*SELECT Namn
FROM person
WHERE ID=29*



Datafil

11 Sven Karlsson Vägen 3 Kalmar	6 Eva Brun Gatan 23 Växjö	17 Ulla Snäll Stigen 1 Nybro	29 Pelle Pall Gränden 3 Kalmar	13 Stina Prat Vägen 34 Visby	37 Kalle Fjäder Kanalgatan 3 Karlskrona
--	------------------------------------	---------------------------------------	---	---------------------------------------	--



PK / FK /Sammansatt?

KundID	Namn	TelID	KundID	Tel
1	Kalle	1	1	123123
2	Otto	2	2	456456
3	Anna	3	4	988654
4	Stina	4	7	234341
5	Sture	5	1	23456
		6	3	1245466
		7	1	234987

Pk TelID
Fk KundID
Tvingande

KundID	Namn	TelID	KundID	Tel
1	Kalle	1	1	123123
2	Otto	2	2	456456
3	Anna	3	4	988654
4	Stina	4	7	234341
5	Sture	5	1	23456
		6	3	1245466
		7	1	234987

Sammansatt
Pk TelID & KundID
Sammansatt=*Beroende*

KundID	Namn	KundID	TelID	Tel
1	Kalle	1	1	123123
2	Otto	1	5	23456
3	Anna	1	7	234987
4	Stina	2	2	456456
5	Sture	3	6	1245466
		4	3	988654
		7	4	234341

Sammansatt
Pk KundID & TelID
Sammansatt=*Beroende*



Nycklar

Namn	Beskrivning
Primary Key	Primär nyckel, unik. Den primära sorteringsordningen. Är alltid indexerat.
Alternate Key Candidate Key	Alternativ primärnyckel. Alla övriga fält som skulle kunna vara Pk.
Foreign Key	Främmande nyckel. Kan vara unik. Nyckel som används för att koppla samma två tabeller. Bör vara indexerat vid många poster. Ej index i MSSQL, Index i MySQL.
Index	Anger en sortering har lagts på ett eller flera fält en tabell. Kan ofta sättas som unikt. Används vanligen för att förbättra åtkomsten vid sökning.
Secondary Key	Sekundär nyckel. Normalt avses det som ett samlingsnamn för andra än Pk.



Ett index medför:

- ✓ Att mer data lagras

Databasen blir större

- ✓ Insert, update och delete tar längre tid

Index måste uppdateras samtidigt som data ändras/läggs till

- ✓ Det går snabbare visa en sorterad lista om det finns index
(`SELECT Enamn, Fnamn FROM Kund ORDER BY Enamn, Fnamn`)

- ✓ Det går snabbare att hitta en post i stora tabeller om det finns index
(`SELECT Enamn WHERE Enamn='Berg'`)

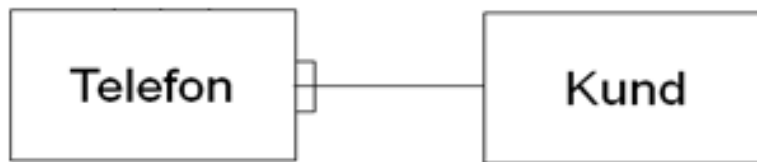
- ✓ Relaterade poster hittas lättare om den främmande nyckeln indexeras

Främmande nycklar indexeras med fördel. (`SELECT ... JOIN... ON yID=xID`)

- ✓ Primärnycklar och **unika** kolumner indexeras vanligen automatiskt



RI handlar om hur man definierar tre olika händelser mellan relaterade tabeller, dvs relationen och då hur data ska behandlas mellan förälder (parent) och barn (child).



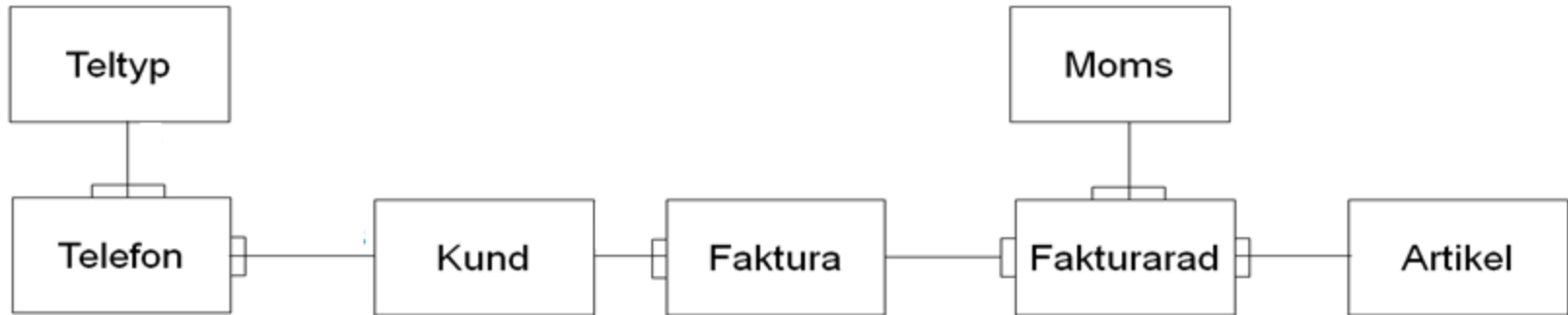
1. INSERT, dvs när nya poster läggs till.
Ska en barnpost (telefon) kunna skrivas in utan att det anges vilken förälder som äger den.
2. DELETE, radering av poster.
Om en förälder raderas ska barnposterna raderas eller ska de vara kvar?
3. UPDATE, ändring av poster
Om ändring av Pk på föräldersida – vad ska då hända med barnposterna?



Begrepp	Beskrivning
No Action	En förälderpost kan inte ändras/raderas om det finns barnposter.
Restrict	Samma som No Action.
Cascade	En förälderpost kan ändas/raderas även om det finns barnposter. Vid radering raderas också barnposterna. Vid ändring ändras också barnposterna.
Set Default	En förälderpost kan ändas/raderas även om det finns barnposter. Vid ändring/radering ändras barnposternas Fk till värdet som angetts i Default Value.
Set Null	En förälderpost kan ändas/raderas även om det finns barnposter. Vid ändring/radering ändras barnposternas Fk till Null. Fältet måste då tillåtas innehålla Null.



Referentiell Integritet, RI (3 av 3)



Tabeller	Relationstyp	RI	Delete	Update
Teltyp – telefon	1:n			
Kund – Telefon	1:n			
Kund – Faktura	1:n			
Faktura – Fakturarad	1:n			
Artikel – Fakturarad	1:n			
Moms – Fakturarad	1:n			

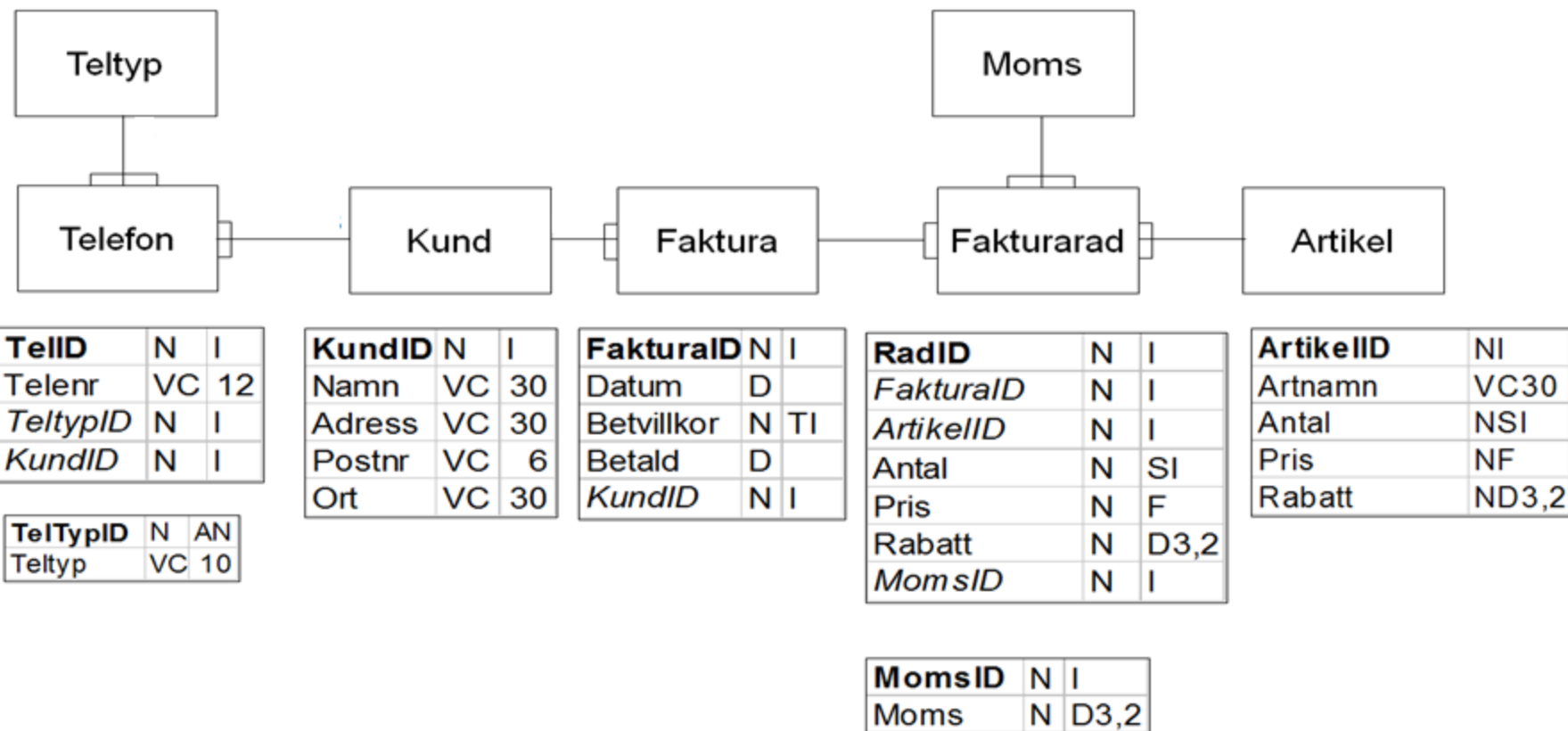
Bestäm hur RI ska vara för din databas.



Fysiska Modellen

Den fysiska modellen ligger som ett direkt underlag till hur du bygger upp databasen. Alla tabeller, fält med fältnamn, datatyper, Pk, Fk, index, beroenden etc.

Din databas får inte skilja sig från din fysiska modell – de ska vara exakt lika.



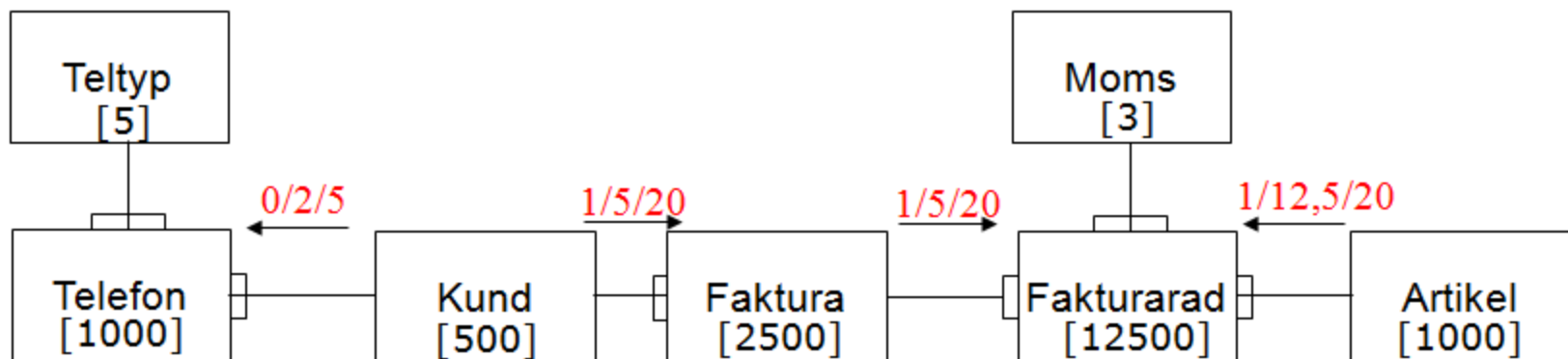


Volymberäkning

Volymberäkning genomförs av flera anledningar:

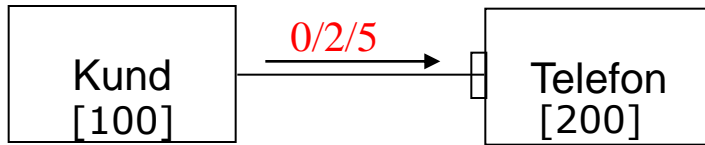
- 1 Att identifiera utrymmesbehovet på lagringsenheter.
- 2 Att identifiera de stora tabellerna som kan ge kapacitetsproblem (belastningsproblem).
- 3 Att identifiera hur vi ska hantera data för historik/statistik
- 4 Att identifiera hur många rader som formulären ska ha för “sina underrader” dvs de rader som ligger på mångasidan i en relation.

- Utför volymberäkningen på den fysiska modellen
- Börja med att ange talen för alla självständiga objekt
- Ta sedan 1:n relationen
- Ta relationsobjekten sist





Volymberäkning



- 1 Ange i de självständiga objekten antalet poster efter 1:a året.
100 kunder har vi i vårt exempel.
- 2 Rita pilen från 1 till mångasidan.
- 3 Ange relationstalen på relationslinjen **(0/2/5)**
0 är lägsta antalet telefoner på en kund
2 är det normala antalet telefoner för en kund
5 är det maximala antalet telefoner för en kund
- 4 Beräkna mångasidan. Normalvärdet multiplicerat med antal kunder ger antalet rader i Telefon. $2 * 100 = 200$ poster i telefon
- 5 För typtabeller anges endast antalet rader direkt (självständiga objekt)
- 6 Vissa objekt råkar ut för en baklängesberäkning

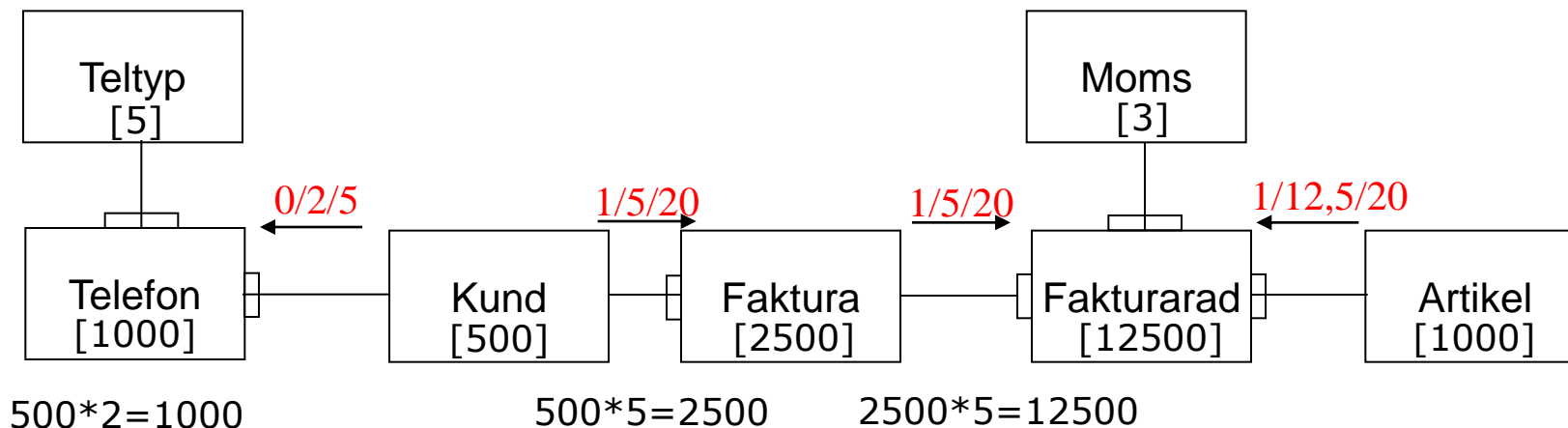
Namn	<input type="text" value="Evensson AB"/>
Adress	<input type="text" value="Ringvägen 12"/>
Postnr	<input type="text" value="39352"/>
Ort	<input type="text" value="KALMAR"/>
Rabatt	<input type="text" value="15"/>
	<input type="text" value="070 5476123"/>
	<input type="text" value="0480-986545"/>
	<input type="text" value="0480-123456"/>
	<input type="text" value="070-987654"/>
	<input type="text" value="0485-987623"/>



Volymberäkning per objekt

Antalet poster i självständiga objekt och typtabeller är angivna

Lägg in relationstal o riktning mellan 1:n relationer. Beräkna!



Relationstalet mellan Artikel och Fakturarad beräknas ur :

Antalet fakturarader delat med antalet artiklar. Dvs $10000/1000$ så får du 10. Normalvärdet är 10.

Lägsta och högsta värde antar du och får exempelvis: $1/10/20$

Nu har du antalet poster som kommer att ingå i tabellerna efter första året och kan beräkna utrymmesbehovet. Du kan också bestämma hur formulären ska se ut för mångasidorna.



Volymberäkning – Tillväxt under fem år

För in värden från det första året i nedanstående tabell och beräkna tillväxten i antalet poster de kommande fyra åren så du får en femårsplan.

- Typtabeller behåller samma antal poster.
- Kundunderlaget ökar med 10% per år. Vilket ökar telefon, fakturor, fakturarader.
- Förra årets fakturor/fakturarader finns kvar i databasen för att statistikbearbetning etc.
- Artikel ökar också med 10%

Tabell	Antaltecken	----- Antal poster -----					---- Diskbehov ----	
		År 1	År 2	År 3	År 4	År 5	År 1	År 5
Teletyp		5	5	5	5	5	0	0
Moms		3	3	3	3	3	0	0
Kund		500	550	605	666	732	0	0
Artikel		1000	1100	1210	1331	1464	0	0
Telefon		1000	1100	1210	1331	1464	0	0
Faktura		2500	5250	8275	11603	15263	0	0
Fakturarad		12500	26250	41375	58013	76314	0	0
							0	0

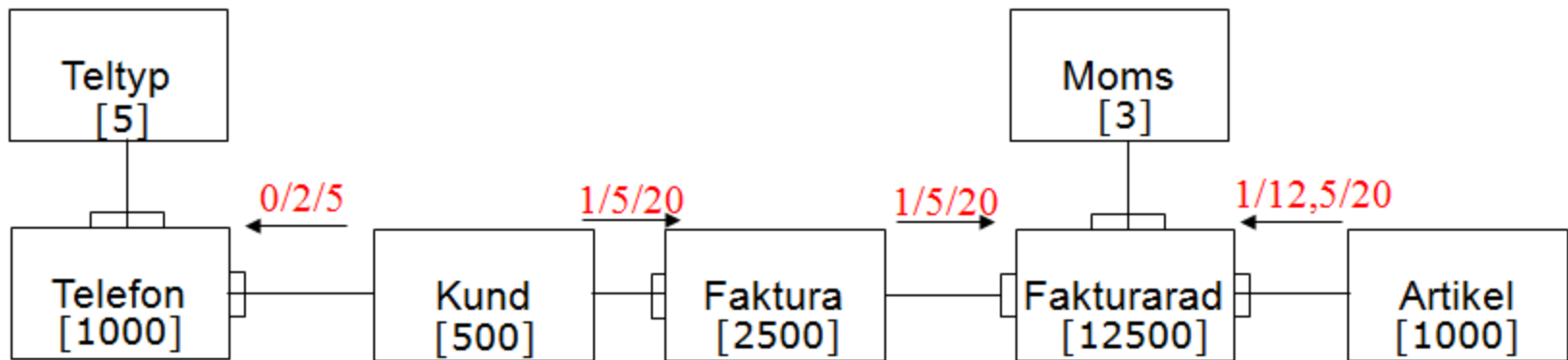


Volymberäkning - utrymmesbehov

TeTypID	INT	4	FakturaID	INT	4
Teltyp	VC 10	10	Datum	Date	3
		14	Betvillkor	TINYINT	1
			Betald	Date	3
TeIID	INT	4	<i>KundID</i>	INT	4
Telenr	VC 15	15	Index på KundID		4
<i>TeTypID</i>	INT	4			19
<i>KundID</i>	INT	4			
Index - 2st INT (fk)		8	FakturaradID	INT	4
		35	<i>FakturaID</i>	INT	4
			<i>ArtikelID</i>	INT	4
KundID	<i>INT</i>	4	Antal	SMALLINT	2
Namn	VC 30	30	Pris	FLOAT	4
Adress	VC 30	30	Rabatt	DECIMAL(3,2)	5
Postnr	VC 6	6	<i>MomsID</i>	INT	4
Ort	VC 25	25	Index 3 st INT		12
Epost	VC 50	50			39
Index på namn		30			
		175	ArtikelID	INT	4
			Artikelnamn	VC30	30
MomsID	INT	4	Antal	SMALLINT	2
Moms	DECIMAL(3,2)	5	Pris	FLOAT	4
		9			40



Volymberäkning - Fakturamodellen

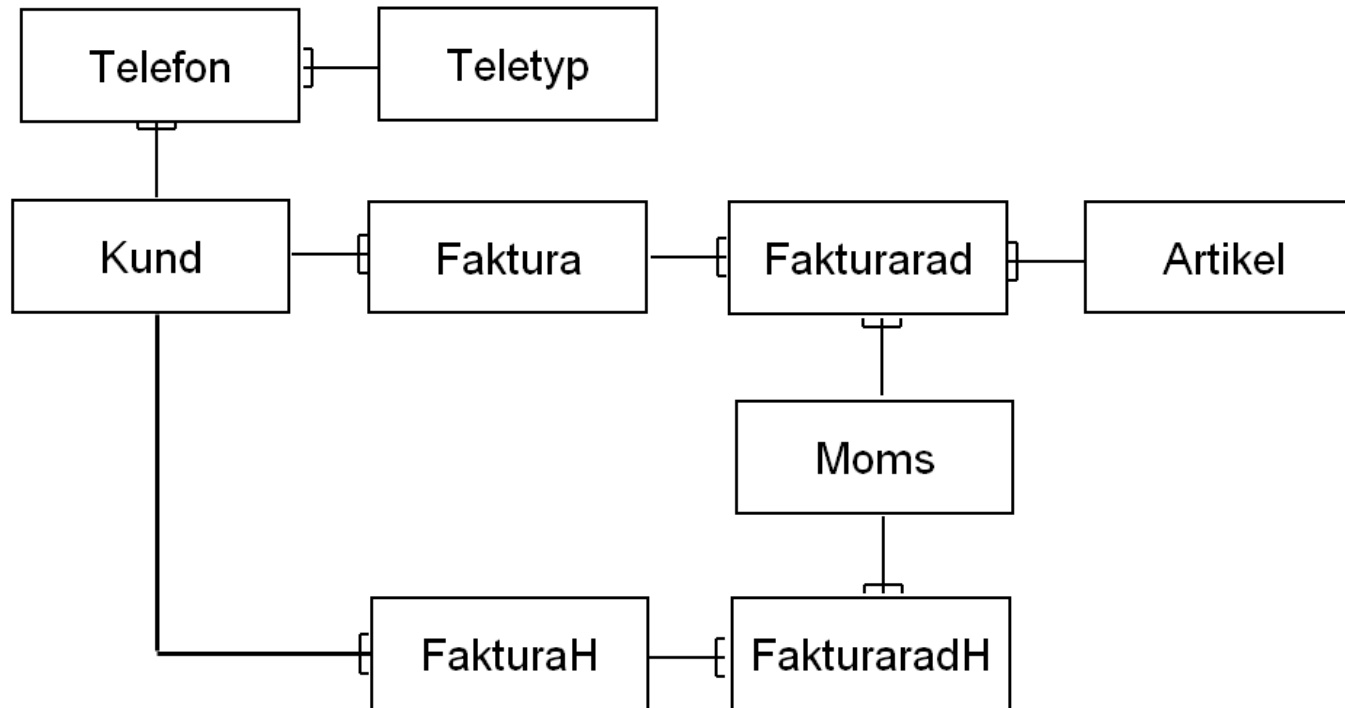


Tabell	Antaltecken	----- Antal poster -----					---- Diskbehov ----	
		År 1	År 2	År 3	År 4	År 5	År 1	År 5
Teletyp	14	5	5	5	5	5	70	70
Moms	9	3	3	3	3	3	27	27
Kund	175	500	550	605	666	732	87 500	128 109
Artikel	40	1 000	1 100	1 210	1 331	1 464	40 000	58 564
Telefon	25	1 000	1 100	1 210	1 331	1 464	25 000	36 603
Faktura	19	2 500	5 250	8 275	11 603	15 263	47 500	289 992
Fakturarad	39	12 500	26 250	41 375	58 013	76 314	487 500	2 976 236
							687 597	3 489 601

Siffrorna visar att Faktura och Fakturarad ökar en hel del. Efter fem år så finns en hel del gamla data som ligger i dessa tabeller och kommer att påverka våra svarstider. Vi skapar därför två stycken tabeller som är precis lika som Faktura o Fakturarad. Inför varje nytt år tar vi gamla data, ex de som är äldre än två år och flyttar över dem till historiktabeller. Vi döper de nya tabellerna till FakturaH och FakturaradH



Historik



Detta blir vår datamodell. Det kommer inte att påverka vårt utrymmesbehov eftersom det fortfarande är samma mängd med data. Vi kommer att ha kvar gamla data och kan köra statistikbearbetning av dessa. Förutsättningen är när vi kör statistik på både gamla o nya data är att vi slår samman Faktura med FakturaH på något sätt och även Fakturarad med FakturaradH. Se exempelvis SQL kommandot UNION där vi kan slå samma data från flera tabeller.