



Modul DB1-1

Databasmodellering

Antal föreläsningar: **2**

Antal laborationer: **1**

Förkunskapskrav: **Databasintroduktion**

Kurslitteratur: **"Praktisk datamodellering"**
ISBN: 91-44-38001-1

Referenslitteratur:

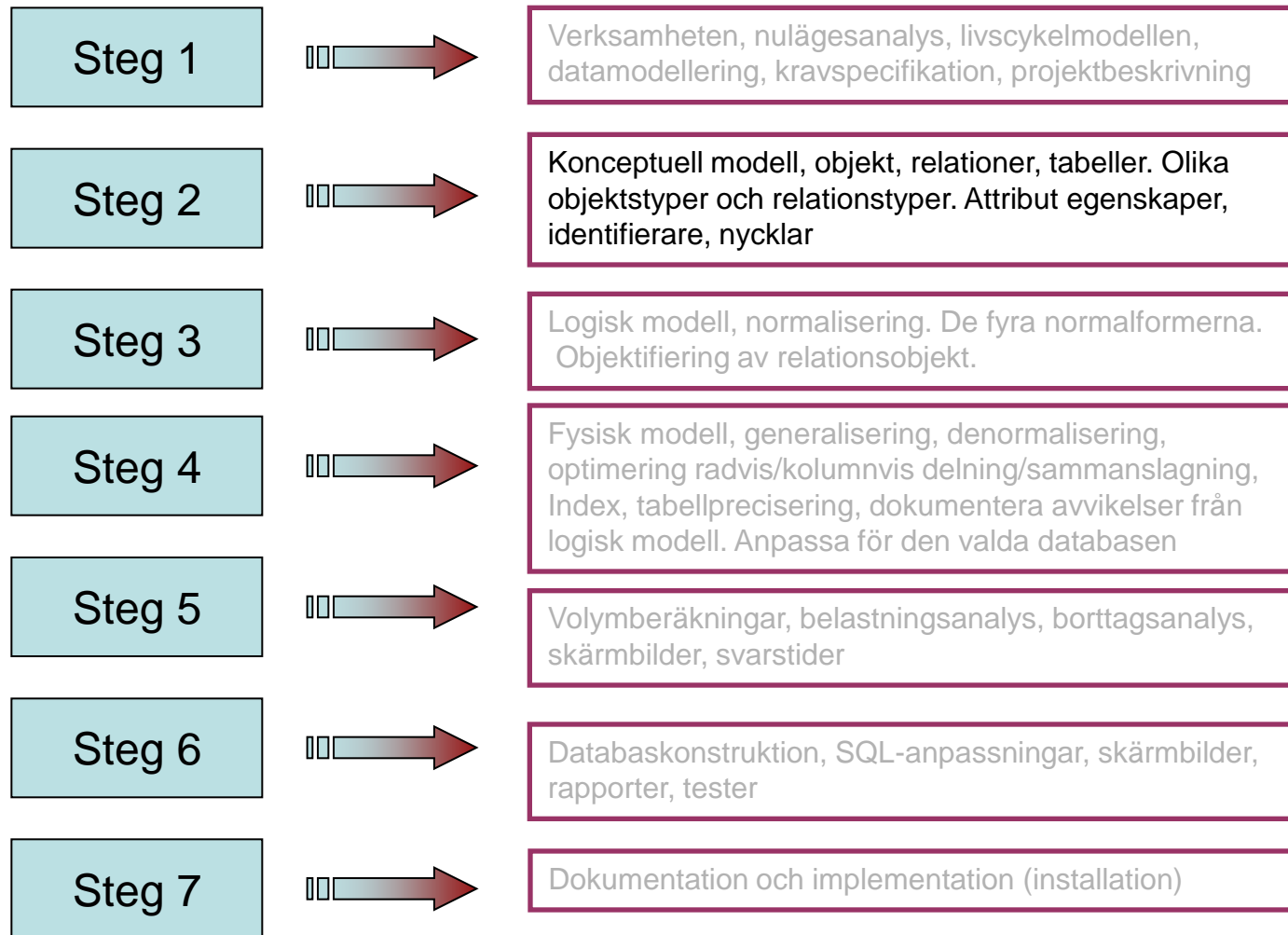


Innehållsförteckning:

- Utvecklingsprocessen
- Databasmodellering
- Min- och maxreglerna
- Tabeller
- Primära och sekundära nycklar
- Relationer
- Bilköpsexempel



Utvecklingsprocessen





Vad är datamodellering?

Datamodellering:

- gör det enklare att beskriva, pröva och enas om hur begrepp och regler i en verksamhet bör utformas.
- Förenklar utveckling och implementation av databaser.

Vad är en bra databasstruktur?

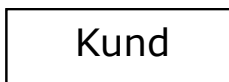
- en databasstruktur som är lätt att skala i storleksordning.
- en databasstruktur som inte dubbellagrar data.
- en databasstruktur som inte är onödigt stor eller komplicerad.
- en databasstruktur som är flexibel och inte innefattar onödiga begränsningar.



Datamodellering – några symboler



Objekt i verksamheten som troligtvis kommer att bli en tabell i databasen. Ritats som en rektangel.



Objektet anges med ett namn. Kund är viktigt objekt för säljverksamheten.

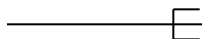


Relation. Mellan två objekt finns en relation. Objekten knyts ihop med en relationslinje

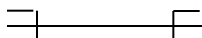
Det finns tre typer av relationer:



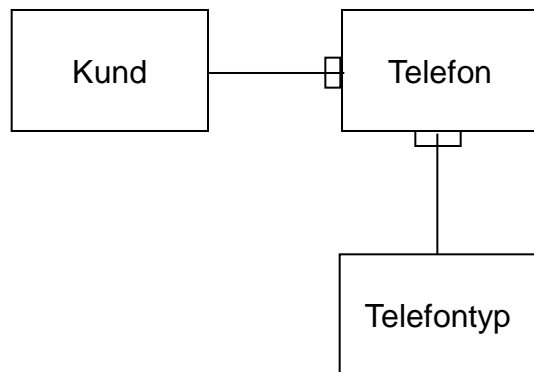
1:1 ett till ett



1:n ett till många och många till ett



n:m många till många

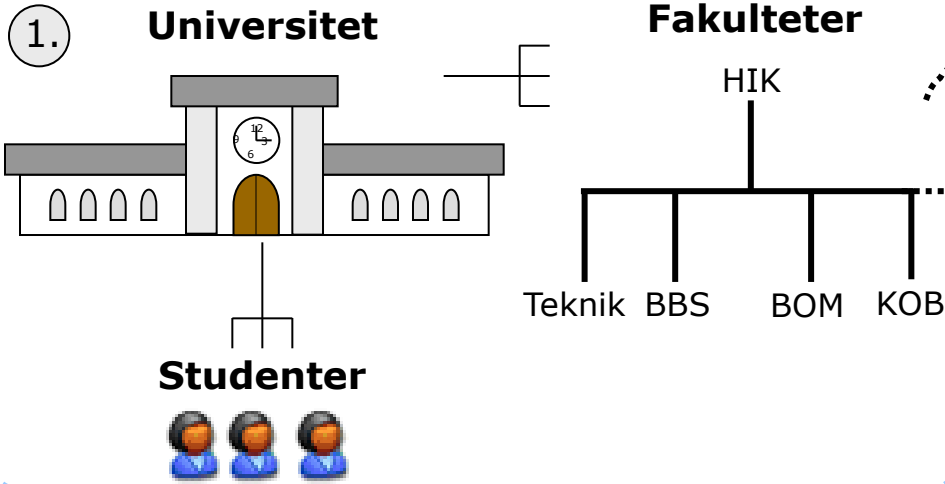


En Kund kan ha flera telefoner. Mellan Kund och Telefon råder en till många. En Telefon tillhör en Kund.

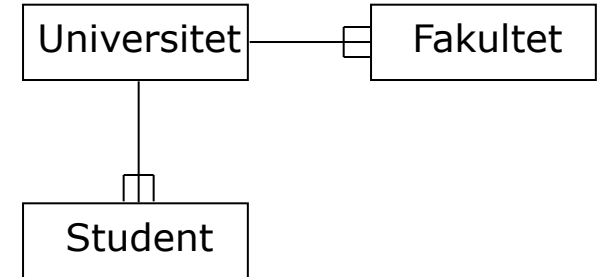
En Telefon är av en viss typ, Telefontyp. En viss Telefontyp finns för en eller flera telefonnummer. (Ex Mobil, Hem, Arbete).

Datamodellering

Verkligheten



Datamodell



Tabeller i databasen

5.

FakID	FakultetNamn	Program	UniID
1	Teknik	Di:p	1
2	BBS	SysVet	1

UniID	Universitetnamn	Adress
1	HIK	Norrvägen 47
2	KTH	Karlavägen 1

StuID	Enamn	Fnamn	UniID
1	Karl	Ljung	1
2	Lotta	Flinta	2

1. Identifiera objekt, attribut och relationer
2. Avbilda verkligheten
3. Rita en datamodell
4. Skapa tabeller
5. Fyll i tabellerna med relevant data för några poster i respektive tabell.

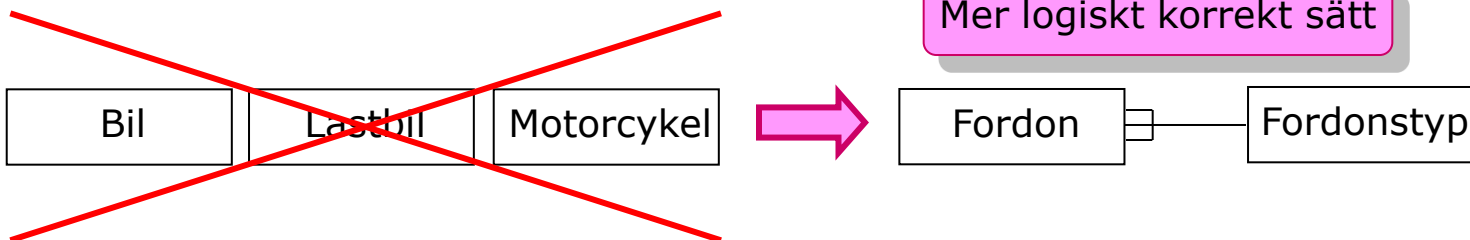


Namngivning av objekt

Varje objekt måste ha ett eget unikt namn.

Resor	Resa
Kunder	Kund
Artiklar	Artikel

Ange namnen i singularis.
Använd inga blanksteg i namnen.
Se till att namnen beskriver innehållet i objektet.
Försök att generalisera genom att använda exempelvis fordon istället för bil, motorcykel eller lastbil.
Om du har bil, lastbil etc som du vill registrera. Använd då ett fält som anger vad det är för typ av Fordon.



Använd gärna det engelska alfabetet a-z och A-Z när du namnger objekten (läs tabellerna) då SQL-språket inte känner till svenska bokstäver. Det går att använda svenska tecken men då måste du skriva [] runt varje namn när du ställer SQL-frågor till databasen.

De slutliga villkoren för namnsättning (när vi översätter till tabellnamn), antal tecken etc, bestäms av den specifika databashanteraren.



Namngivning av fält i tabeller

**Varje fält i en tabell
måste ha ett eget
uniktt namn.**

Ange namnen i singularis.
Använd inga blanksteg i namnen.
Se till att namnen beskriver innehållet i
fältet.

ID	Förnamn	Gatu adress
1	Otto	Venusvägen 1
2	Anita	Storgatan 23

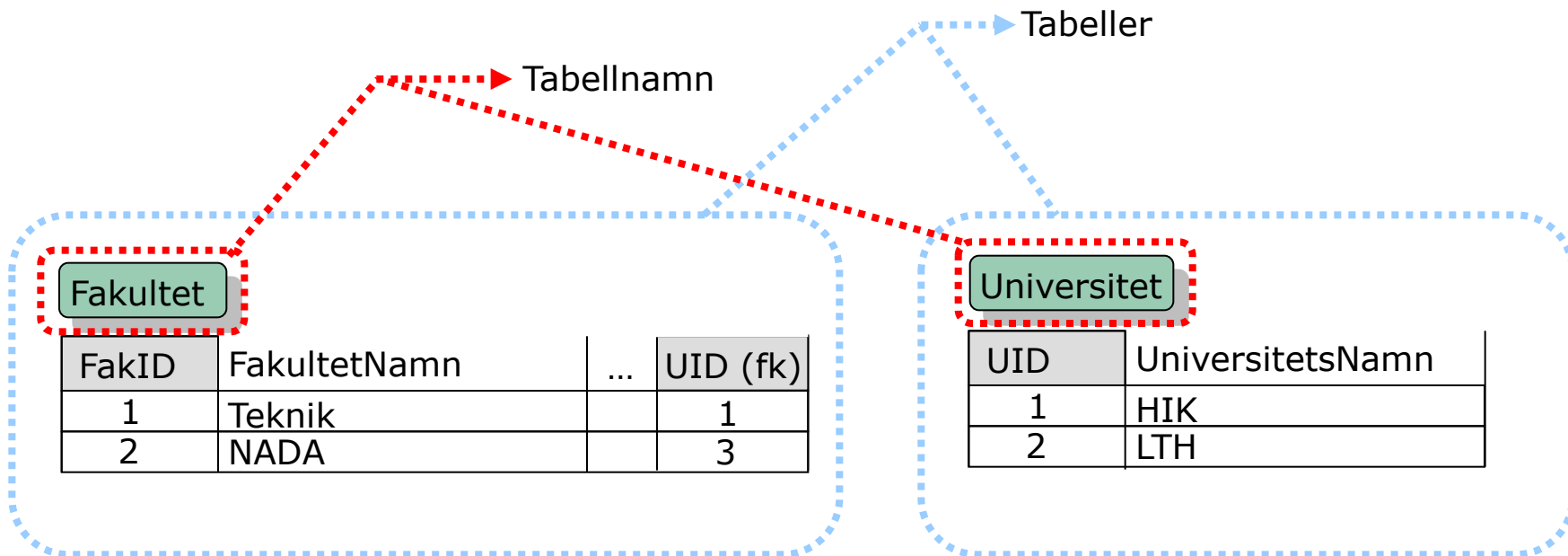
ID	Fnamn	Adress
1	Otto	Venusvägen 1
2	Anita	Storgatan 23

Använd gärna det engelska alfabetet a-z och A-Z när du namnger objekten (läs tabellerna) då SQL-språket inte känner till svenska bokstäver. Det går att använda svenska tecken men då måste du skriva [] runt varje namn när du ställer SQL-frågor till databasen.

De slutliga villkoren för namnsättning, antal tecken etc, bestäms av den specifika databashanteraren.



Tabeller





Tabeller forts.

Kolumnnamn,
egenskapsnamn eller
fältnamn

Fakultet

FakID	FakultetNamn	...	UID (fk)
1	Teknik		1
2	NADA		3

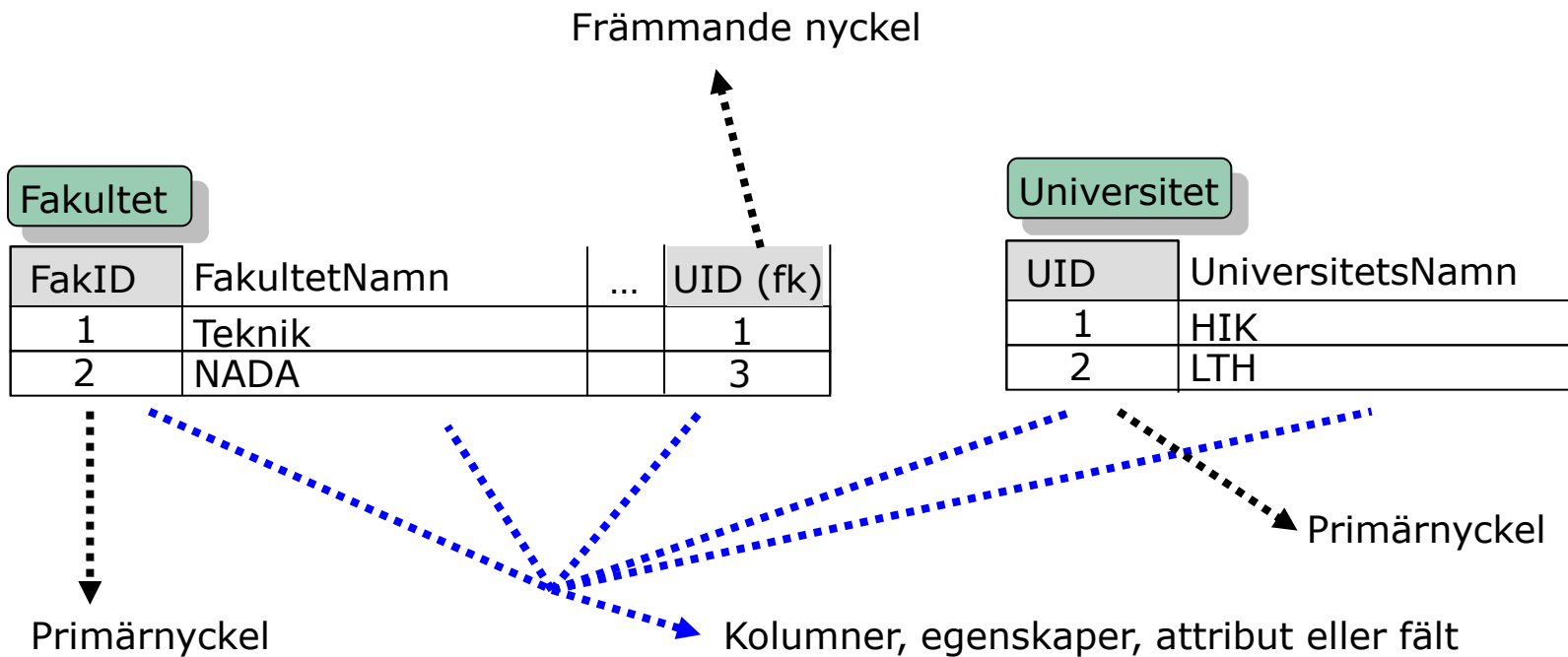
Universitet

UID	UniversitetsNamn
1	HIK
2	LTH

Rader, förekomster eller poster



Tabeller forts.





Primärnyckel


Primärnyckel (Primary Key) förkortas ofta som pk

Huvudsyftet med en primärnyckel är att unikt identifiera en post i databasen

- Består av ett eller flera fält i en tabell.
- Väljs av lämpliga existerande fält i tabellen eller ett nyskapat fält för primärnyckeln.
- Nyckeln måste vara stabil då det är kostsamt och kräver en stor arbetsinsats för att byta namn på den vid ett senare tillfälle.
- Undvik att använda så kallade talande fält som primärnyckel.

Exempelvis: (Talande fält => fält som **kanske** ändras med tiden)

- Artikelnummer.
- Personnummer.
- Namn.
- Autonummerade nycklar är att föredra (IDENTITY i MS SQL-Server, AUTO_INCREMENT i MySQL, Autonumber (Räknare) i MS Access).
- Markeras med ett tak i tabellen.
- Numeriska nycklar ger bättre prestanda.
- Indexeras automatiskt i de flesta databaser.



PID	Namn
1	Kalle
2	Lotta

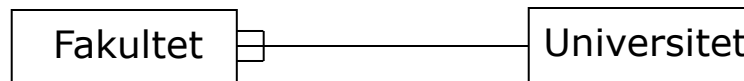


Främmande nyckel

Främmande nyckel (Foreign key) förkortas ofta som fk

- Gör det möjligt att sammankoppla tabellerna med varandra.
- Den främmande nyckeln utgörs av innehållet i primärnyckel från den relaterade tabellen.
- Markeras utan tak i tabellen.

Exempelvis:

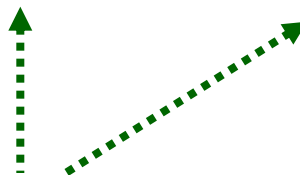


Fakultet

FakID	FakultetsNamn	...	UID (fk)
1	Teknik		1
2	NADA		3

Universitet

UID	UniversitetsNamn
1	HIK
2	LTH



Minnesregel:

- Du skapar ett nytt fält som har samma namn (helst) och datatyp som primärnyckeln i den relaterade tabellen på ett-sidan.



De olika objekttyperna

– Självständiga objekt

- Oberoende av andra objekt.
- Har en egen unik primärnyckel.

Exempelvis:

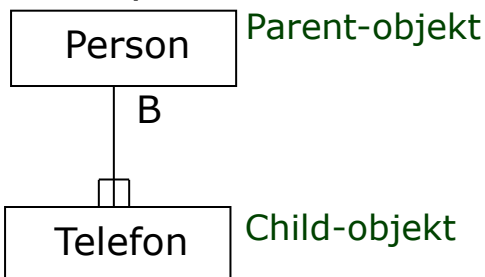
Produkt

ProdID	Beskrivning	Antal	...
1	Skruv	50	
2	Mutter	200	

– Beroendeobjekt

- Ägs av ett eller flera överordnade objekt.
- Primärnyckeln är sammansatt av ägarens nyckel och det underliggande objektets nyckel.

Exempelvis:



PID	TelID	Telefonnummer	Typ
17	1	08-56875536	Hem
17	2	070-7845765	Mobil
18	1	0480-565274	Hem

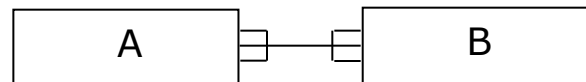
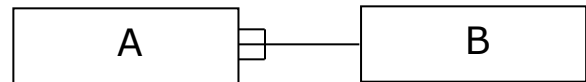
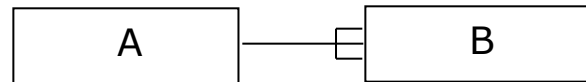
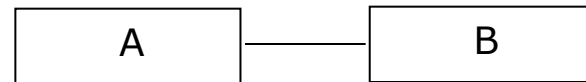
Registrering av telefonnummer sker normalt inte utan ägare



Relationer

- Sammanbandet mellan de olika tabellerna
- En relation kan ha olika kardinalitet, relationstyper

- 1 - 1 (en till en relation)
- 1 - n (en till många relation)
- n - 1 (Många till en relation)
- n - m (Många till många relation)



- Kardinaliteten bestäms utifrån en förekomst av det objekt som du börjar läsa ifrån.

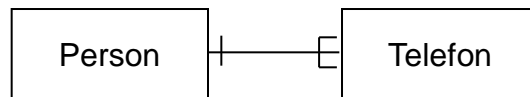
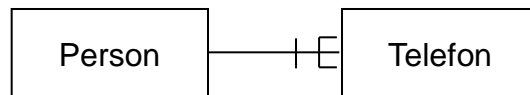
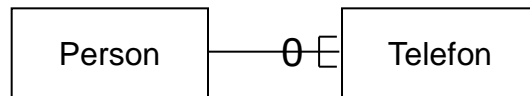
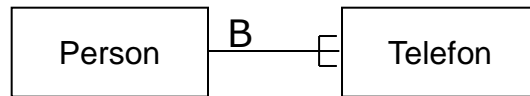
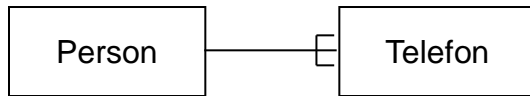


Min- och Maxreglerna

Symboler			Max	Min	Typ av relation
<u>1..*</u>	<u>1..n</u>	— E	Många	1	Tvingande
	<u>1</u>	—	1	1	Tvingande
<u>0..*</u>	<u>0..n</u>	—E	Många	0	Frivillig
		—0—	1	0	Frivillig
		—	1	0	Frivillig



Min- och Maxreglerna / exempel



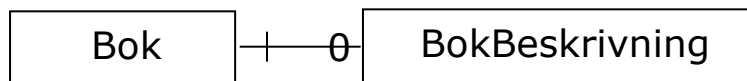
- Frivillig. Fk måste inte anges i Telefon. Därmed kan ett telefonnummer existera utan någon relation på 1-sidan
- **B** anger att Telefon är beroende av Person. Ett B ger en sammansatt nyckel på många sidan. Personid måste ingå i sammansatt nyckel på mångasidan.
- Frivillig i Telefonnummer. När en person registreras måste inte telefonnummer anges. Markeras normalt inte.
- Tvingande. När en person registreras måste ett telefonnummer också läggas in.
- Tvingande i Person. En Person måste registreras om en Telefon ska registreras. Sammansatt nyckel krävs inte. Se också B för Beroende.



En- till en-relation (1 - 1)

- Prestandaskäl
 - Onödigt att belasta en tabell med för stora poster om du inte alltid använder alla fält.
- Säkerhetsskäl
 - Lättare att skydda en tabell än enstaka kolumner i en tabell.
- Används för kompletterande information som gäller enstaka poster
 - För att inte få tomma fält i databasen.

Exempelvis:



Bok

BokID	BokNamn	...
10	SQL-programmering	
20	SQL-Administrering	

- Koppling mellan primärnyckel och primärnyckel
- Har samma ID i en 1 - 1 relation

BokBeskrivning

BeskID	BokBeskrivning	...
20	Gul framsida	

Bok + BokBeskrivning

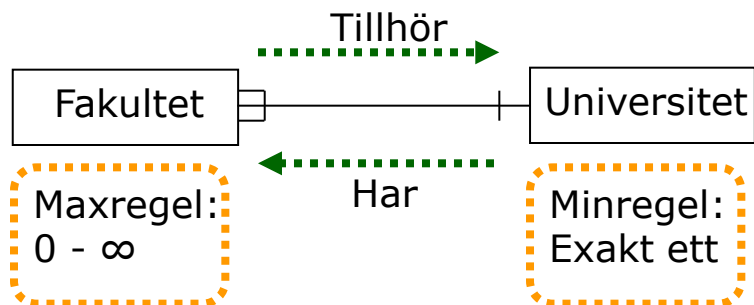
BokID	BokNamn	...	BokBeskrivning
10	SQL-programmering		
20	SQL-Administrering		Gul framsida

←..... Tomt fält



1:n och n:1 relationer

En till många ⇔ många till en (1:n, n:1)



Fakultet

FakID	FakultetNamn	...	UID (fk)
1	Teknik		1
2	NADA		3

Universitet

UID	UniversitetsNamn
1	HIK
2	LTH
3	KTH

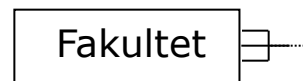
Du kan läsa datamodeller på följande sätt:



En fakultet tillhör exakt ETT Universitet



En fakultet kan existera utan ett Universitet



Ett Universitet har många fakulteter, ner till inga fakulteter



Ett Universitet har många fakulteter, ner till EN fakultet

Primärnyckeln från ett sidan blir främmande nyckel på många sidan

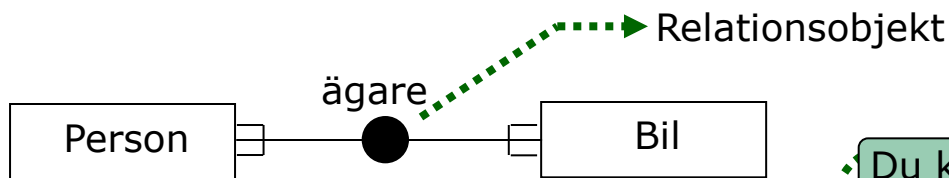


Varför inte lägga Universitetsnamnet direkt i fakultetstabellen?

- För att undvika redundans (dubbelagring), samma Universitetsnamn skulle lagras flera gånger.
- När/Om ett Universitetsnamn ska ändras, behöver du bara ändra på ett ställe för att det ska gälla för alla poster med det namnet.
- Universitetsnamnen kan skrivas in även om det inte finns några fakulteter.
- Du kan lätt sortera på Universitetsnamn.
- Du får samma stavning på Universitetsnamnen och du undviker på så sätt problem när du ska jämföra Universitetsnamn mot varandra.



Många till många relationer (n:m)



Läsordning:

1. →
2. ←

Du kan läsa datamodellen på följande sätt:

1. En person kan äga en eller flera bilar
2. En bil kan ägas av en eller flera personer

Tabellen "ägare"

- Ägare kallas relationsobjekt eftersom det realiseras med en egen tabell vars pk är sammansatt av pk från person- och biltabellerna

Ägare

PID	BilID
100	10
200	20
300	30

Ägare som kopplingstabell



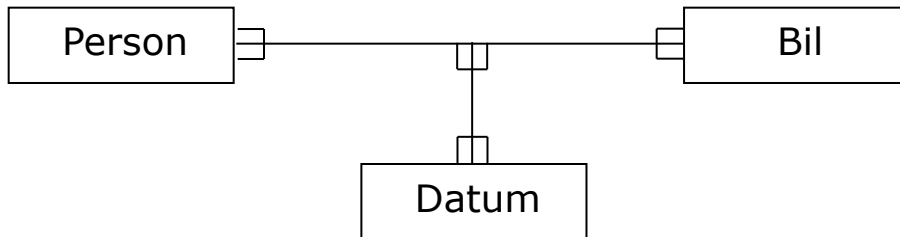
Bilköpet

Exempel:

Så vill vi att det ska fungera:

En **person** köper en eller flera **bilar** vid olika **datum**
En bil kan köpas av flera personer vid olika datum
Ett visst datum kan det ske ett eller flera bilköp

1. Identifiera vilka de olika objekten är (Person, datum, bil)
2. Skapa en konceptuell datamodell med de identifierade dataobjekten som uppfyller de kraven vi ställde ovan



3. Eftersom vi får många till många relationer måste vi använda oss av relationsobjekt som kopplingstabeller mellan de olika tabellerna.
 - Var ska relationsobjektet placeras i den konceptuella datamodellen?

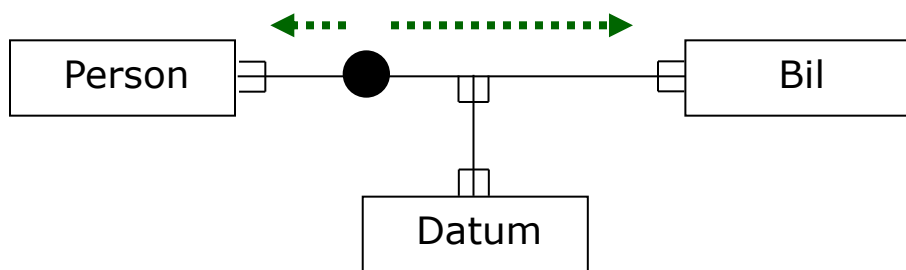


Bilköp forts.

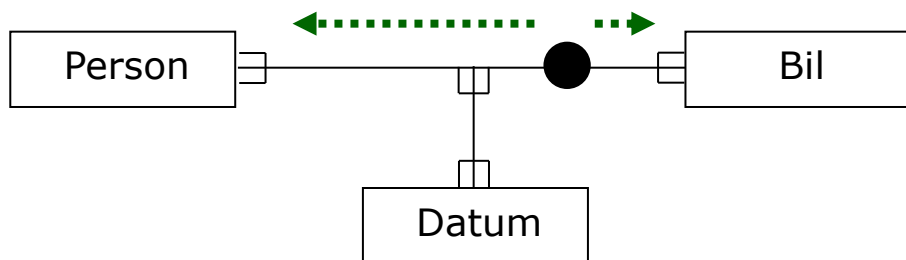
4.

Vi kan placera relationsobjektet på tre olika platser i den konceptuella datamodellen

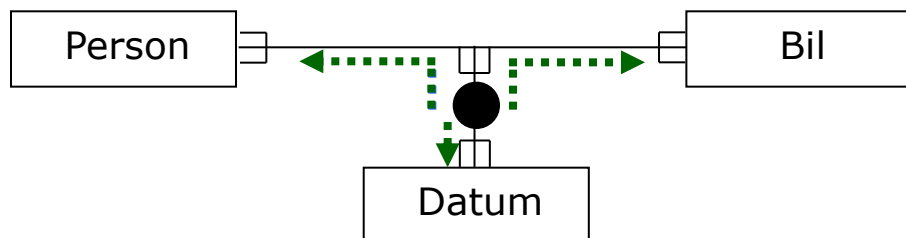
I.



II.



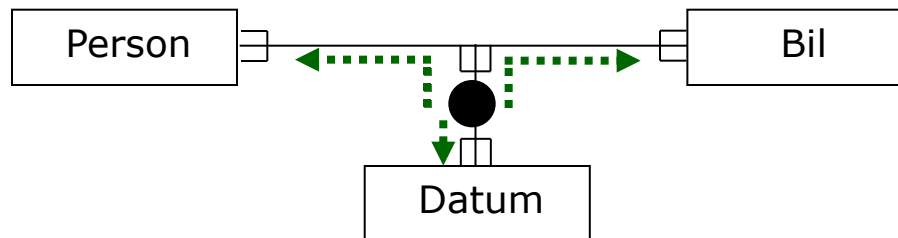
III.



Alternativ tre är det enda alternativet som uppfyller våra krav då vi där även får med datumet



Bilköp forts.



Person				
PID	Namn	Adress	Postnr	Ort
1	Kalle Ek	Stigen 23	393 51	KALMAR
2	Stor Björk	Boken 12	394 63	KALMAR

Datum	
DID	Datum
1	2006-03-15
2	2006-03-16

Bil			
BID	Märke	Typ	Regnr
1	Volvo	S60	ABC123
2	Volvo	V50	CBA321

Köp		
PID	BID	DID
1	2	1
1	1	1

Relationsobjektet får sin sammansatta Pk från omgivande objekt.
I relationsobjektet visas att Kalle Ek har köpt en Volvo V50 den 15/3 2006 och att samma Kalle Ek har också köpt en Volvo S60 den 15/3 2006.

En **person** köper en eller flera **bilar** vid olika **datum**
En bil kan köpas av flera personer vid olika datum
Ett visst datum kan det ske ett eller flera bilköp