

SQL Introduksjonskurs

Oversikt



Oversikt

- Introduksjon til datamodellering
- Normalisering
- Logisk skjema til Database
- Strukturelle operasjoner
- Operasjoner mot data
- Kontrolloperasjoner
- Aggregering og indekser
- Analysere et fiktiv fagsystem

Hva skal vi jobbe med

- Vi skal primært jobbe med og utvikle en Noark 5 struktur
 - Inneholder de fleste elementene vi forventer å bevare
 - Bygger på XML kurset
- Gjennom kurset skal vi
 - Definere strukturen til en Noark 5 base
 - Sette data inn i Noark 5 basen
 - Utvikle spørringer mot Noark 5 basen
- Helheten med det vi jobber med kommer først imorgen

Hvordan skal vi jobbe

- MySQL Workbench
- Utvikle database kommandoer for å forstå hva vi kan gjøre med en database
- Alt vi gjør her kan kjøres fra en MySQL terminal
 - Du er ikke avhengig av MySQL Workbench for å bruke det du lærer
- Vi skal også lagre alle kommandoene vi bruker i en tekst fil
 - Notepad++ anbefales, men notepad fungerer også

Del 1

Tid	Tema	Beskrivelse
10.10 – 10:55	Datamodellering og normalisering	Hvordan kommer vi fram til at en bestemt database struktur er riktig? ER modellering introduseres. Er det konsekvenser med datalagring som vi må være oppmerksom på? Data lagret i tabeller kan føre til innsetting, oppdatering og slette anomalier. Normalisering introduseres for å begrense disse anomaliene.
11.10 – 12.00	Logisk skjema og nøkler	Vi ser på hvordan en ER modell kan brukes som grunnlag til en faktisk implementasjon av en database struktur. Primær og fremmed nøkler introduseres og er forklart. Sekundær nøkler

Del 2

Tid	Tema	Beskrivelse
12:45 – 13:45	Strukturelle operasjoner	Data Definition Language (DDL) brukes for å definere/redigere struktur i en database. Dette gjelder både for tabeller og koloner. Her vil vi se på følgende kommandoer: CREATE, ALTER, DROP, TRUNCATE
14.00 – 14.50	Strukturelle operasjone	Fortsetter fra siste time
15.05 – 16.00	Operasjoner mot data	Data Manipulation Language (DML) brukes for å administrere data. Vi ser på følgende kommandoer: INSERT, DELETE, UPDATE
16.10 – 17.00	Kontrolloperasjoner	Data Control Language (DCL) brukes for å gi og frata rettigheter fra en bruker

Del 3

Tid	Tema	Beskrivelse
09:00 – 09:50	SQL, SQL og aggregering	Vi begynner med å forstå SQL syntaks og ser hvordan SQL kan brukes til å filtrere og aggregere data
10.00 – 10.50	SQL og Joins	
11.05 – 12.00	Indekser, Views og eksport	

7/171



Del 4

Tid	Tema	Beskrivelse
12.45 – 13:45	Gjennomgangstime	Vi bruker denne timen for å ta opp temaer som kanskje er fortsatt litt uklart.
14.00 – 14.50	Praktisk arbeid	De to siste timene brukes for å jobbe opp mot en fiktiv fagsystem som dere får tilgang til. Dere skal analysere basen og svare på en del spørsmål om basen som dere må løse med SQL
15.05 – 15.50	Praktisk arbeid	Fortsetter arbeidet fra siste time

8/171



SQL Introduksjonskurs

Datamodellering, Normalisering og Logisk skjema



Oversikt

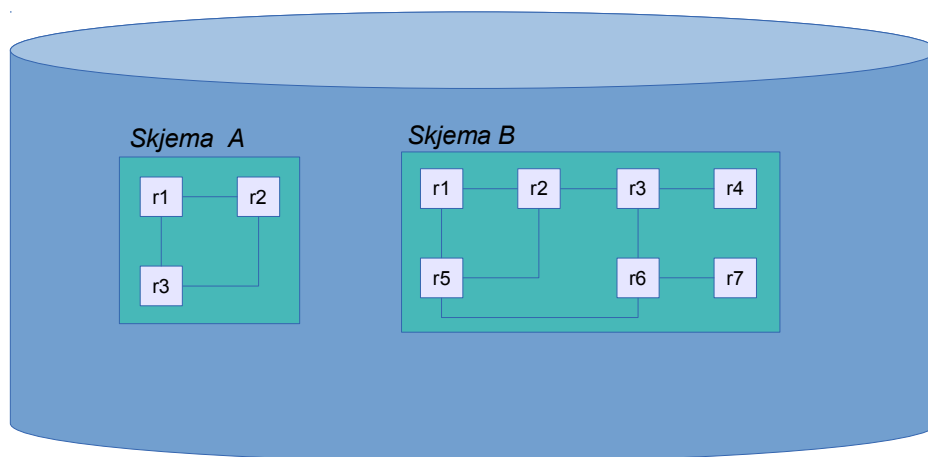
- Introduksjon til datamodellering
- Normalisering
- Logisk skjema til Database

Relasjonsmodellen

- En database kalles et skjema
- Data er lagret i relasjoner* (tabeller)
- Tilgang til data er (vanligvis) med nøkler
- To sentrale nøkkeltyper brukt i relasjoner
 - Primærnøkkel
 - Fremmednøkkel

*I en tabell kan du ha duplikate rader, ikke lov i en relasjon

DBHS



En databasehåndteringsystem kan bestå av mange skjemaer. Ofte kalles en *skjema* bare for en *database* eller *base*

Data lagret i relasjoner (tabeller)

Bil				
RegistreringsNr	UnderstellsNr	Farge	Produsent	Modell
LH12984	10946534	Rød	Volkswagen	Golf
DK23491	9648573	Blå	Toyota	Yaris
BP12349	5523840	Grønn	Skoda	Fabia
ZT97495	2643923	Hvit	Seat	Leon

Relasjon, Attributter, Tuppler

1 Relasjon



Biler				
RegistreringsNr	UnderstellsNr	Farge	Produsent	Modell
LH12984	10946534	Rød	Volkswagen	Golf
DK23491	9648573	Blå	Toyota	Yaris
BP12349	5523840	Grønn	Skoda	Fabia
ZT97495	2643923	Hvit	Seat	Leon

←←←← *4 Tuppler*



5 Attributter

Grovt sett ...

En relasjon er en tabell*

EtAttributt er en kolonne

En tuppel er en rad

**I en tabell kan du duplikate rader, ikke lov i en relasjon*

Primærnøkler

- En primærnøkkel er en verdi som kan brukes til å identifisere en unik rad i en relasjon
- Primærnøkkelen identifiserer et unikt objekt fra et sett av objekter
 - Personnummer som identifiserer et spesifikk menneske
 - Skiltnummer på en bil
- På engelsk heter dette *Primary Key*

Fremmednøkkel

- En fremmednøkkel er et felt i en tabell i en relasjonsdatabase som peker til et felt i (vanligvis) en annen tabell
- Dette siste feltet er ofte tabellens primærnøkkel
- På denne måten kan tabeller kobles sammen
- På engelsk heter dette *Foreign Key*

Primær- og Fremmed Nøkler

Student		
<u>StudentNr</u>	Fornavn	Etternavn
12345	Jan	Karlsen
23456	Pål	Solberg
34567	Mette	Johansen
45678	Ingrid	Aleksandersen

StudentTelefonNr	
<u>StudentNr</u>	TelefonNr
12345	76543829
23456	90783298
34567	99456543
45678	45990234

StudentNr er primærnøkkel i begge relasjonene

Men **StudentNr** i **Student** relasjonen er en fremmednøkkel til **StudentTelefonNr** relasjonen

Primær- og Fremmed Nøkler

Konto	
KontoNr	Saldo
15486110	8,000
06584585	2,000
95486110	8,000
06759425	-3,000

Kunder		
KunderNr	Navn	Etternavn
1	Pål	Solberg
2	Nils	Nilsen
3	Ari	Hansen

KontoEier	
KundeNr	KontoNr
1	15486110
1	06584585
2	15486110
2	95486110
3	95486110
3	06759425

- **KontoNr** er primærnøkkel i **Konto** relasjonen og fremmednøkkel til **KontoEier** relasjonen
- **KundeNr** er primærnøkkel i **Kunder** relasjonen og fremmednøkkel til **KontoEier** relasjonen
- **KundeNr** og **KontoNr** er primærnøkler i **KontoEier** relasjonen

Lagring av data

- Kan vi bare lagre data eller er det noe vi må ta høyde for?
- Redundans og anomalier
 - Innsetting
 - Oppdatering
 - Sletting
- Vi jobber med en oppdiktet scenario
 - Du har en liten utleiefirma der du leier 3-4 biler og registrerer alt i Excel
 - Bruker dette eksempel for å problematisere data modellering

Redundans

- Redundans betyr at dataene dine gjentar seg og gjør basen din unødvendig stor og dette kan forårsake feil med dataene

Nr	Navn	Alder	Førekort	Adresse	Telefon	Bil	Produsent	Modell	Drivstoff	Farge	Leifra	Leietil	Km stand
1	Hans Hansen	39	B	Osloveien 1 0140 Oslo	89564578 / 32154879	ZK34567	Volkswagen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	12-01-2012	15-01-2012	900
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grønn	15-01-2012	18-01-2012	100500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
2	Jens Jensen	78	A/B/C	Solveien 23 1371 Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
6	Michael Michaelsen	45	B	Drammensveien 1 0001 Oslo	45856954	ZK34567	Volkswagen	Golf	Diesel	Grønn	19-01-2012	20-01-2012	100900

21/171



Innsettingsanomali

- Hver gang en kunde leier en bil så må all data om kunden og bilen settes inn på nytt.
- Hvis bil informasjon er påbudt, kan vi ikke sette inn data om en kunde uten at faktisk de leier en bil
- Hvis kunde informasjon er påbudt, kan vi ikke sette inn data om en bil uten at faktisk det er koblet mot en kunde

Nr	Navn	Alder	Førekort	Adresse	Telefon	Bil	Produsent	Modell	Drivstoff	Farge	Leifra	Leietil	Km stand
1	Hans Hansen	39	B	Osloveien 1 0140 Oslo	89564578 / 32154879	ZK34567	Volkswagen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	12-01-2012	15-01-2012	900
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grønn	15-01-2012	18-01-2012	100500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
2	Jens Jensen	78	A/B/C	Solveien 23 1371 Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
6	Michael Michaelsen	45	B	Drammensveien 1 0001 Oslo	45856954	ZK34567	Volkswagen	Golf	Diesel	Grønn	19-01-2012	20-01-2012	100900
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	22-01-2012	23-01-2012	1100
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grønn	23-01-2012	24-01-2012	101500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	24-01-2012	25-01-2012	1800

22/171



Oppdateringsanomali

- Hvis en bil blir lagt inn med feil farge og må senere oppdateres må systemet først finne alle relevante forekomster
- Hvis ikke alle forekomster blir funnet og endret så har vi inkonsistens i dataene

Original data

Nr	Navn	Alder	Førekort	Adresse	Telefon	Bil	Produsent	Modell	Drivstoff	Farge	Leiefra	Leietil	Km stand
1	Hans Hansen	39	B	Osloveien 1 0140 Oslo	89564578 / 32154879	ZK34567	Volkswagen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	12-01-2012	15-01-2012	900
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grønn	15-01-2012	18-01-2012	100500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
2	Jens Jensen	78	A/B/C	Solveien 23 1371 Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
6	Michael Michaelsen	45	B	Drammensveien 1 0001 Oslo	45856954	ZK34567	Volkswagen	Golf	Diesel	Grønn	19-01-2012	20-01-2012	100900
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	22-01-2012	23-01-2012	1100
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grønn	23-01-2012	24-01-2012	101500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	24-01-2012	25-01-2012	1800

Oppdatert data

Nr	Navn	Alder	Førekort	Adresse	Telefon	Bil	Produsent	Modell	Drivstoff	Farge	Leiefra	Leietil	Km stand
1	Hans Hansen	39	B	Osloveien 1 0140 Oslo	89564578 / 32154879	ZK34567	Volkswagen	Golf	Diesel	Grå	12-01-2012	14-01-2012	100000
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	12-01-2012	15-01-2012	900
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grå	15-01-2012	18-01-2012	100500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
2	Jens Jensen	78	A/B/C	Solveien 23 1371 Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
6	Michael Michaelsen	45	B	Drammensveien 1 0001 Oslo	45856954	ZK34567	Volkswagen	Golf	Diesel	Grå	19-01-2012	20-01-2012	100900
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	22-01-2012	23-01-2012	1100
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grå	23-01-2012	24-01-2012	101500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	24-01-2012	25-01-2012	1800

Sletteanomali

- Hvis en kunde leier en ny bil og så avbestiller den kan all informasjon om bilen bli borte

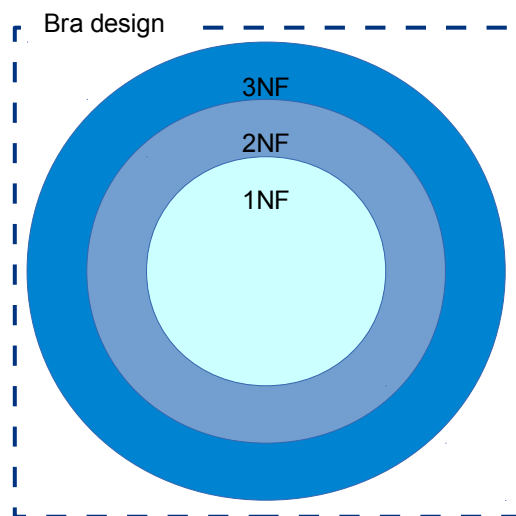
Nr	Navn	Alder	Førekort	Adresse	Telefon	Bil	Produsent	Modell	Drivstoff	Farge	Leiefra	Leietil	Km stand
1	Hans Hansen	39	B	Osloveien 1 0140 Oslo	89564578 / 32154879	ZK34567	Volkswagen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	12-01-2012	15-01-2012	900
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grønn	15-01-2012	18-01-2012	100500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
2	Jens Jensen	78	A/B/C	Solveien 23 1371 Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
6	Michael Michaelsen	45	B	Drammensveien 1 0001 Oslo	45856954	ZK34567	Volkswagen	Golf	Diesel	Grønn	19-01-2012	20-01-2012	100900
7	Tor Torisen	52	B	Finnmarksveien 29 0142 Oslo	72138598	EL10000	Tesla	Model S	Strøm	Rød	19-01-2012	20-01-2012	0

Nr	Navn	Alder	Førekort	Adresse	Telefon	Bil	Produsent	Modell	Drivstoff	Farge	Leiefra	Leietil	Km stand
1	Hans Hansen	39	B	Osloveien 1 0140 Oslo	89564578 / 32154879	ZK34567	Volkswagen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	12-01-2012	15-01-2012	900
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grønn	15-01-2012	18-01-2012	100500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
2	Jens Jensen	78	A/B/C	Solveien 23 1371 Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
6	Michael Michaelsen	45	B	Drammensveien 1 0001 Oslo	45856954	ZK34567	Volkswagen	Golf	Diesel	Grønn	19-01-2012	20-01-2012	100900

Normalisering

- Metode som brukes for å kontrollere om man har modellert en god database struktur
- Hvorfor gjør vi normalisering?
 - Minimalisere duplisering av data
 - Under oppdatering av data må systemet være konsekvent og integritet må ivaretaes
 - Forhindre at anomaliene vi så på oppstår
- Når gjør vi normalisering?
 - Starter tidlig i database design prosessen

Normalformene



Første Normalformen

En tabell er i første normalform hvis og bare hvis alle koloner kun inneholder atomære verdier

Atomære verdier betyr at hver rute i tabellen kan inneholde kun en verdi

27/171



Første Normalformen

Nr	Navn	Alder	Førerkort	Adresse	Telefon	Bil	Produsent	Modell	Drivstoff	Farge	Leiefra	Leietil	Km stand
1	Hans Hansen	39	B	Solvøien 1 0140 Oslo	89564578 / 32154879	ZK34567	Volkswagen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
2	Jens Jensen	78	A/B/C	Drammensvein 38 1371 Asker	12457898 / 36965478	DK12457	Peugeot	306	Bensin	Svart	12-01-2012	15-01-2012	900
3	Maria Mariasen	54	B	Askerveien 45 1430 Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grønn	15-01-2012	18-01-2012	100500
4	Marte Martinsen	23	C/D	Suppegjøkveien 32 1431 Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
2	Jens Jensen	78	A/B/C	Solvøien 23 1371 Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
6	Michael Michaelsen	45	B	Drammensveien 1 0001 Oslo	45856954	ZK34567	Volkswagen	Golf	Diesel	Grønn	19-01-2012	20-01-2012	100900

Nr	Fnavn	Enavn	Alder	Førerkort	Adresse	PNM	Sted	Telefon	Bil	Produsent	Modell	Drivstoff	Farge	Leiefra	Leietil	Km stand
1	Hans	Hansen	39	B	Oslovøien 1	0140	Oslo	89564578	ZK34567	Volkswagen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
1	Hans	Hansen	39	B	Oslovøien 1	0140	Oslo	32154879	ZK34567	Volkswagen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
2	Jens	Jensen	78	A	Drammensvein 38	1371	Asker	12457898	DK12457	Peugeot	306	Bensin	Svart	12-01-2012	15-01-2012	900
2	Jens	Jensen	78	A	Drammensvein 38	1371	Asker	36965478	DK12458	Peugeot	306	Bensin	Svart	12-01-2011	15-01-2011	900
2	Jens	Jensen	78	B	Drammensvein 38	1371	Asker	12457898	DK12459	Peugeot	306	Bensin	Svart	12-01-2010	15-01-2010	900
2	Jens	Jensen	78	B	Drammensvein 38	1371	Asker	36965478	DK12460	Peugeot	306	Bensin	Svart	12-01-2009	15-01-2009	900
2	Jens	Jensen	78	C	Drammensvein 38	1371	Asker	12457898	DK12461	Peugeot	306	Bensin	Svart	12-01-2008	15-01-2008	900
2	Jens	Jensen	78	C	Drammensvein 38	1371	Asker	36965478	DK12462	Peugeot	306	Bensin	Svart	12-01-2007	15-01-2007	900
3	Maria	Mariasen	54	B	Askerveien 45	1430	Ås	65987445	ZK34567	Volkswagen	Golf	Diesel	Grønn	15-01-2012	18-01-2012	100500
4	Marte	Martinsen	23	C	Suppegjøkveien 32	1431	Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
4	Marte	Martinsen	23	D	Suppegjøkveien 32	1431	Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
2	Jens	Jensen	78	A	Solvøien 23	1371	Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
2	Jens	Jensen	78	B	Solvøien 23	1371	Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
2	Jens	Jensen	78	C	Solvøien 23	1371	Asker	87964585	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
6	Michael	Michaelsen	45	B	Drammensveien 1	0001	Oslo	45856954	ZK34567	Volkswagen	Golf	Diesel	Grønn	19-01-2012	20-01-2012	100900

28/171

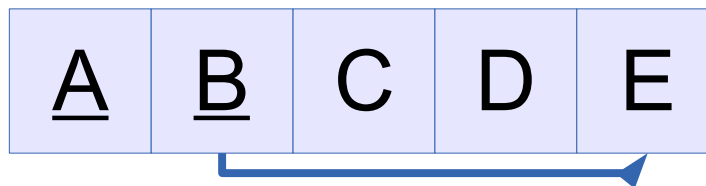


Andre Normalformen (2)

En tabell er i andre normalform (2NF) hvis og bare hvis den er i 1NF og alle koloner som ikke er en del av primærnøkkelen er funksjonelt avhengige av hele primærnøkkelen, og ikke bare deler av den

Brudd på 2NF

A og B er primærnøkler



Avhengighet mellom B og E

Identifisere Primærnøkkel

- Før vi går videre, må vi identifisere primærnøkklene
- Husk primærnøkkelen er en unik nøkkel som identifiserer en unik rad i en relasjon i en database
- En primærnøkkel kan bestå av 1 eller flere koloner men du kan aldri ha same primærnøkkel over to rader

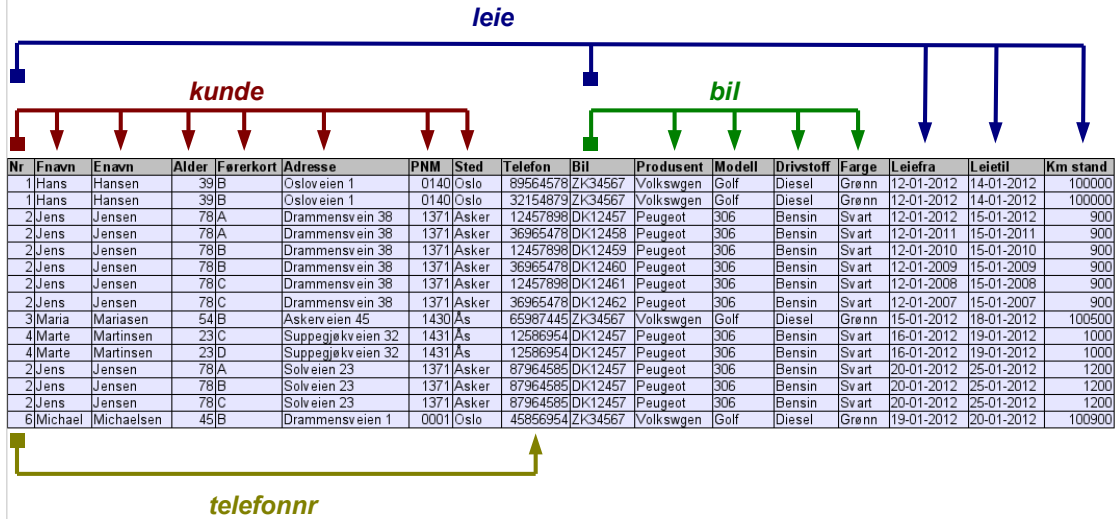
Identifiser Primærnøkler

Nr	Enavn	Enavn	Alder	Førekort	Adresse	PNM	Sted	Telefon	Bil	Produsent	Modell	Drivstoff	Farge	Leifra	Leietil	Km stand
1	Hans	Hansen	39	B	Osloveien 1	0140	Oslo	89564578	ZK34567	Volkswgen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
1	Hans	Hansen	39	B	Osloveien 1	0140	Oslo	32154879	ZK34567	Volkswgen	Golf	Diesel	Grønn	12-01-2012	14-01-2012	100000
2	Jens	Jensen	78	A	Drammensvein 38	1371	Asker	12457898	DK12457	Peugeot	306	Bensin	Svart	12-01-2012	15-01-2012	900
2	Jens	Jensen	78	A	Drammensvein 38	1371	Asker	36965478	DK12458	Peugeot	306	Bensin	Svart	12-01-2011	15-01-2011	900
2	Jens	Jensen	78	B	Drammensvein 38	1371	Asker	12457898	DK12459	Peugeot	306	Bensin	Svart	12-01-2010	15-01-2010	900
2	Jens	Jensen	78	B	Drammensvein 38	1371	Asker	36965478	DK12460	Peugeot	306	Bensin	Svart	12-01-2009	15-01-2009	900
2	Jens	Jensen	78	C	Drammensvein 38	1371	Asker	12457898	DK12461	Peugeot	306	Bensin	Svart	12-01-2008	15-01-2008	900
2	Jens	Jensen	78	C	Drammensvein 38	1371	Asker	36965478	DK12462	Peugeot	306	Bensin	Svart	12-01-2007	15-01-2007	900
3	Maria	Mariasen	54	B	Askerveien 45	1430	Ås	65987445	ZK34567	Volkswgen	Golf	Diesel	Grønn	15-01-2012	18-01-2012	100500
4	Marte	Martinsen	23	C	Suppegjøkveien 32	1431	Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
4	Marte	Martinsen	23	D	Suppegjøkveien 32	1431	Ås	12586954	DK12457	Peugeot	306	Bensin	Svart	16-01-2012	19-01-2012	1000
2	Jens	Jensen	78	A	Solveien 23	1371	Asker	87964588	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
2	Jens	Jensen	78	B	Solveien 23	1371	Asker	87964588	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
2	Jens	Jensen	78	C	Solveien 23	1371	Asker	87964588	DK12457	Peugeot	306	Bensin	Svart	20-01-2012	25-01-2012	1200
6	Michael	Michaelsen	45	B	Drammensveien 1	0001	Oslo	45856954	ZK34567	Volkswgen	Golf	Diesel	Grønn	19-01-2012	20-01-2012	100900

31/171



Identifiser avhengigheter



32/171



Løsning 2NF

Kunde

Nr	Enavn	Enavn	Alder	Førekort	Adresse	PNM	Sted
1	Hans	Hansen	39	B	Osloveien 1	0140	Oslo
2	Jens	Jensen	78	A	Drammensvein 38	1371	Asker
3	Maria	Mariasen	54	B	Askerveien 45	1430	Ås
4	Marte	Martinsen	23	D	Suppegjøkveien 32	1431	Ås
6	Michael	Michaelsen	45	B	Drammensveien 1	0001	Oslo

Leie

Bil	Nr	Leiefra	Leietil	Km stand
ZK34567	1	12-01-2012	14-01-2012	100000
DK12457	2	12-01-2012	15-01-2012	900
ZK34567	3	15-01-2012	18-01-2012	100500
DK12457	4	16-01-2012	19-01-2012	1000
DK12457	2	20-01-2012	25-01-2012	1200
ZK34567	6	19-01-2012	20-01-2012	100900

Bil

Bil	Produsent	Modell	Drivstoff	Farge
ZK34567	Volkswgen	Golf	Diesel	Grønn
DK12457	Peugeot	306	Bensin	Svart

Telefonnr

Nr	Telefon
1	89564578
1	32154879
2	12457898
2	36965478
3	65987445
4	12586954
6	45866954

Løsningen er å dele tabellen opp i 4 forskjellige tabeller

- Kunde
- Bil
- Leie
- Telefonnr

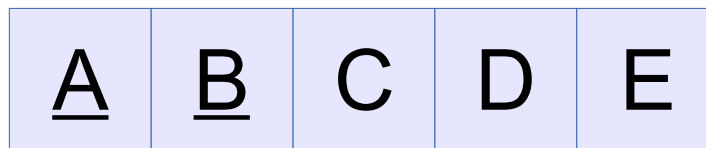
33/171



Tredje Normalformen (3NF)

En tabell er i tredje normalform hvis og bare hvis den er i andre normalform og alle koloner som ikke er en del av primærnøkkelen, er gjensidig uavhengig

Brudd på 3NF
A og B er primærnøkler



Avhengighet mellom C og E

Er alle koloner gjensidig uavhengig?

Kunde

Nr	Fnavn	Enavn	Alder	Førekort	Adresse	PNM	Sted
1	Hans	Hansen	39	B	Osloveien 1	0140	Oslo
2	Jens	Jensen	78	A	Drammensvein 38	1371	Asker
3	Maria	Mariasen	54	B	Askerveien 45	1430	Ås
4	Marte	Martinsen	23	D	Suppegjøkveien 32	1431	Ås
6	Michael	Michaelsen	45	B	Drammensveien 1	0001	Oslo

Leie

Bil	Nr	Leiefra	Leietil	Km stand
ZK34567	1	12-01-2012	14-01-2012	100000
DK12457	2	12-01-2012	15-01-2012	900
ZK34567	3	15-01-2012	18-01-2012	100500
DK12457	4	16-01-2012	19-01-2012	1000
DK12457	2	20-01-2012	25-01-2012	1200
ZK34567	6	19-01-2012	20-01-2012	100900

Bil

Bil	Produsent	Modell	Drivstoff	Farge
ZK34567	Volkswagen	Golf	Diesel	Grønn
DK12457	Peugeot	306	Bensin	Svart

Telefonnr

Nr	Telefon
1	89564578
1	32154879
2	12457898
2	36965478
3	65987445
4	12586954
6	45866954

35/171



Løsning 3NF

Kunde

Nr	Fnavn	Enavn	Alder	Førekort	Adresse	PNM
1	Hans	Hansen	39	B	Osloveien 1	0140
2	Jens	Jensen	78	A	Drammensvein 38	1371
3	Maria	Mariasen	54	B	Askerveien 45	1430
4	Marte	Martinsen	23	D	Suppegjøkveien 32	1431
6	Michael	Michaelsen	45	B	Drammensveien 1	0001

Leie

Bil	Nr	Leiefra	Leietil	Km stand
ZK34567	1	12-01-2012	14-01-2012	100000
DK12457	2	12-01-2012	15-01-2012	900
ZK34567	3	15-01-2012	18-01-2012	100500
DK12457	4	16-01-2012	19-01-2012	1000
DK12457	2	20-01-2012	25-01-2012	1200
ZK34567	6	19-01-2012	20-01-2012	100900

Bil

Bil	Produsent	Modell	Drivstoff	Farge
ZK34567	Volkswagen	Golf	Diesel	Grønn
DK12457	Peugeot	306	Bensin	Svart

Telefonnr

Nr	Telefon
1	89564578
1	32154879
2	12457898
2	36965478
3	65987445
4	12586954
6	45866954

Postnr

PNM	Sted
0140	Oslo
1371	Asker
1430	Ås
1431	Ås
0001	Oslo

36/171



Er det alt vi må tenke på?

- Vi må også se på referanseintegritet mellom tabellene
- Hva skjer i leie tabellen hvis jeg **sletter** en kunde?

Kunde

Nr	Fnavn	Enavn	Alder	Førerkort	Adresse	PNM
1	Hans	Hansen	39	B	Osloveien 1	0140
2	Jans	Jansen	78	A	Drammensvein 38	1321
3	Maria	Mariassen	54	B	Askerveien 45	1430
4	Marte	Martinsen	23	D	Suppegjøveien 32	1431
6	Michael	Michaelsen	45	B	Drammensveien 1	0001

Leie

Bil	Nr	Leiefra	Leietil	Km stand
ZK34567	1	12-01-2012	14-01-2012	100000
DK12457	2	12-01-2012	15-01-2012	300
ZK34567	3	15-01-2012	18-01-2012	100500
DK12457	4	18-01-2012	19-01-2012	1000
DK12457	2	20-01-2012	25-01-2012	1200
ZK34567	6	19-01-2012	20-01-2012	100900

37/171



Referanseintegritet

- Hva skjer i leie tabellen hvis jeg **endrer** en kunde?

Kunde

Nr	Fnavn	Enavn	Alder	Førerkort	Adresse	PNM	Sted
1	Hans	Hansen	39	B	Osloveien 1	0140	Oslo
2	Erna	Ernassen	25	A	Gjøvikveien 32	9001	Gjøvik
3	Maria	Mariassen	54	B	Askerveien 45	1430	Ås
4	Marte	Martinsen	23	D	Suppegjøveien 32	1431	Ås
6	Michael	Michaelsen	45	B	Drammensveien 1	0001	Oslo

Leie

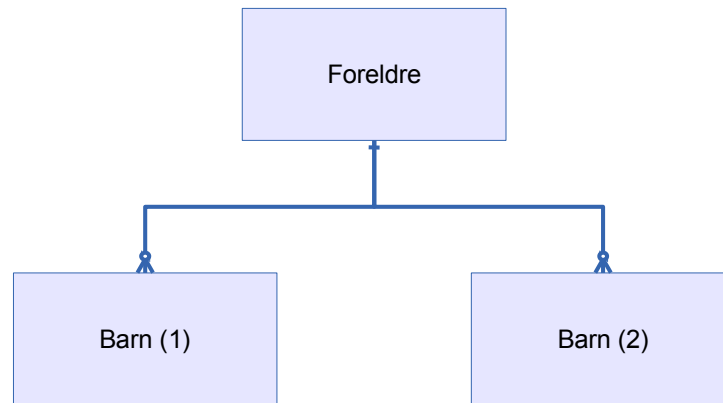
Bil	Nr	Leiefra	Leietil	Km stand
ZK34567	1	12-01-2012	14-01-2012	100000
DK12457	2	12-01-2012	15-01-2012	300
ZK34567	3	15-01-2012	18-01-2012	100500
DK12457	4	18-01-2012	19-01-2012	1000
DK12457	2	20-01-2012	25-01-2012	1200
ZK34567	6	19-01-2012	20-01-2012	100900

38/171



Referanseintegritet

- Håndhevelsen av referanseintegritet går fra 'foreldre' til 'barn'



39/171



Litt mer konkret

- Når referanseintegritet er aktivert gjelder visse regler for dataene
 - Du kan ikke legge til en rad i et barn (tabell) hvis det ikke finnes en tilsvarende verdi i foreldre tabellen (FK)
 - Kan ikke lege til en journalpost uten en sak
 - Du kan ikke slette en rad fra en foreldre (tabell) hvis en tilsvarende rad finnes i et barn tabell
 - Kan ikke slette en sak hvis den har journalposter
 - Du kan ikke endre verdier i primærnøkkel i et foreldre (tabell) hvis det er en relatert rad i et barn (tabell)
 - Kan ikke endre saksnr* hvis saken har journalposter

*saksnr er FK

ER Modelling

- ER Modelling står for «Entity Relationship» eller på norsk «Entitets Sammenhenger»
- Det brukes når du ønsker å utvikle en database som lagrer data for et bestemt scenario
- Det kan brukes til å avgrense og resonere om scenarioet
- Det er en konseptuell modellerings teknikk som definerer entiteter (en del av en scenario, konsepter eller ting) og sammenhengene mellom disse konseptene
- Det er en prosess, som brukes til å skape en datamodell for et gitt problemområde

41/171



ER Modelling

- Konseptuell datamodell uten referanse til teknologi
- Ikke utelukkende for å utvikle databaser, men vi ser på det for database utvikling
- Det kan modellere informasjonsbehovet til en organisasjon eller for et gitt prosjekt
- Sluttproduktet av en datamodellerings prosessen er et relasjons database skjema
- Bruker en Top-down tilnærming
- Veldig nyttig for å forstå databasene sine

42/171



Hva er en entitet?

- En entitet representerer en type (konseptuell) *objekt* ikke en fysisk ting
 - «Bil»
 - ikke «VW Golf» eller «Toyota Yaris»
 - «Lufthavn»
 - ikke, «Gardermoen» eller «Flesland»
 - «Bygning»
 - ikke «P48» eller «P52»
- *Objektet* kan være representert med kun én entitet
- En entitet må være entydig identifiserbar

En entitet



Entitetens navn

Entitetens Navn

Entiteten inneholder attributter

Entitetens Navn
Attributt 1
Attributt 2
Attributt 3
Attributt 4
Attributt 5

“Et attributt er en detalj som kvalifiserer, identifiserer, klassifiserer eller uttrykker tilstanden/status til en enhet”

Eksempler på entiteter

Bil	Flyplass	Bygning
RegistreringsNr	Navn	Navn
UnderstellsNr	Int Kode	Plassering
Farge	Sted	Adresse
Produsent	Land	Postnummer
Modell		Sted

47/171



Entitet Eksempel : Bil

Bil
RegistreringsNr
UnderstellsNr
Farge
Produsent
Modell

RegistreringsNr	UnderstellsNr	Farge	Produsent	Modell
LH12984	10946534	Red	Volkswagen	Golf
DK23491	9648573	Blue	Toyota	Yaris
BP12349	5523840	Green	Skoda	Fabia
ZT97495	2643923	White	Seat	Leon

48/171



Entitet Eksempel : Flyplass

Flyplass
Navn
Int Kode
Sted
Land

Navn	Int Kode	Sted	Land
Gardermoen	OSL	Oslo	Norway
Heathrow	LHR	London	UK
Stanstead	STN	London	UK
Dublin	DUB	Dublin	Ireland

49/171



Entitet Eksempel : Bygning

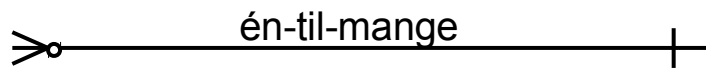
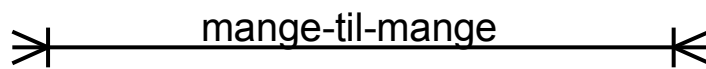
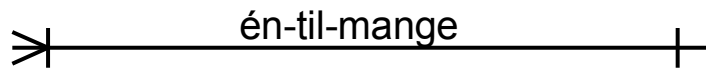
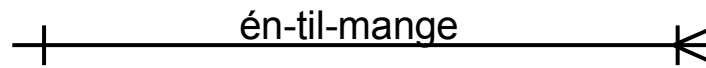
Bygning
Navn
Plassering
Adresse
Postnummer
Sted

Navn	Plassering	Adresse	PostNummer	Sted
P48	Frydenlund	Pilestredet 48	0176	Oslo
P52	Frydenlund	Pilestredet 52	0176	Oslo
Fyrhuset	Frydenlund	Wergelandsv. 27	0176	Oslo
F5	Falbesgate	Falbesgate 5	0170	Oslo

50/171



Sammenhengstyper



Obligatorisk sammenheng



Valgfri sammenheng

Hva betyr én-til-én

En én-til-én sammenheng mellom to tabeller (Tabell A og Tabell B) oppstår når hver rad i *Tabell A* kan kun ha én relatert rad i *Tabell B*

Student		
StudentNr	Fornavn	Etternavn
12345	Jan	Karlson
23456	Pål	Solberg
34567	Mette	Johansen
45678	Ingrid	Aleksandersen

StudentTelefonNr	
StudentNr	TelefonNr
12345	76543829
23456	90783298
34567	99456543
45678	45990234

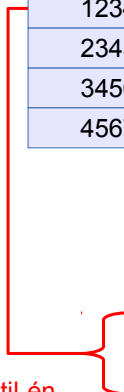
Hva betyr én-til-én

Altså er dette er ulovlig med en én-til-én sammenheng

Student		
StudentNr	Fornavn	Etternavn
12345	Jan	Karlson
23456	Pål	Solberg
34567	Mette	Johansen
45678	Ingrid	Aleksandersen

StudentTelefonNr	
StudentNr	TelefonNr
12345	76543829
12345	90783298
34567	99456543
45678	45990234

Ulovlig i en én-til-én relasjon



Hva betyr én-til-mange

Én-til-mange sammenhenger oppstår når hvert enkelt rad i *Tabell A* kan ha mange relaterte rader i *Tabell B* men hver rad i *Tabell B* har bare en relatert rad i *Tabell A*

Student		
StudentNr	Fornavn	Etternavn
12345	Jan	Karlson
23456	Pål	Solberg
34567	Mette	Johansen
45678	Ingrid	Aleksandersen

StudentTelefonNr	
StudentNr	TelefonNr
12345	76543829
12345	90783298
34567	99456543
34567	45990234



Hva betyr mange-til-mange

Mange-til-mange sammenhenger oppstår når hver rad i *Tabell A* kan ha mange relaterte rader i *Tabell B* og hver rad i *Tabell B* kan ha mange relaterte rader i *Tabell A*

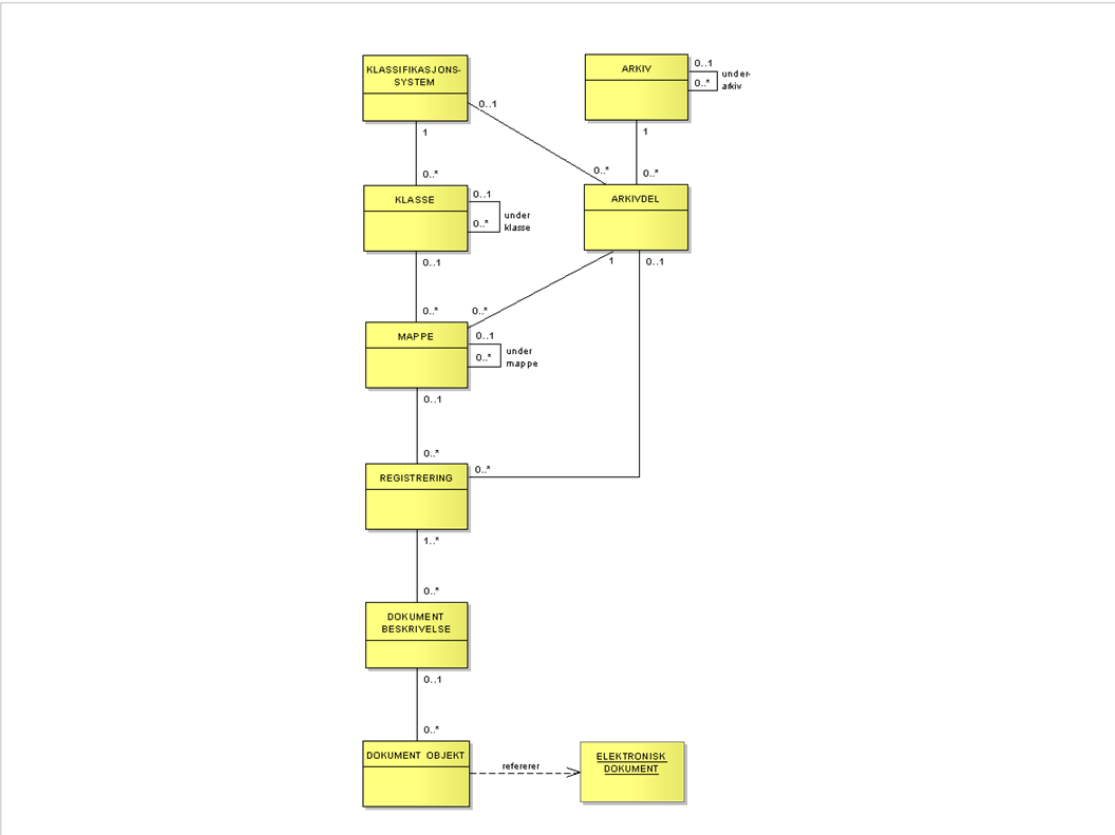
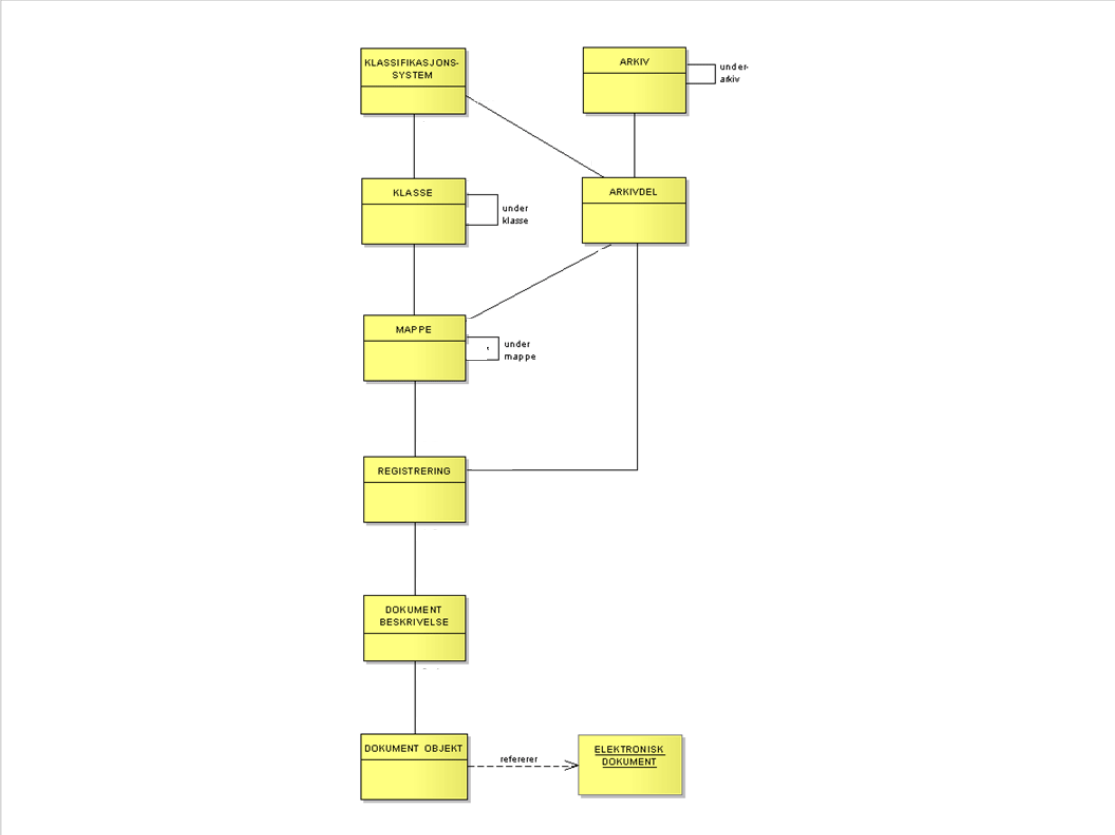
Student		
<u>StudentNr</u>	Fornavn	Etternavn
12345	Jan	Karlson
12345	Jan	Karlson
34567	Mette	Johansen
45678	Ingrid	Aleksandersen

dette er ikke mulig

StudentTelefonNr	
<u>StudentNr</u>	<u>TelefonNr</u>
12345	76543829
12345	90783298
34567	99456543
34567	45990234

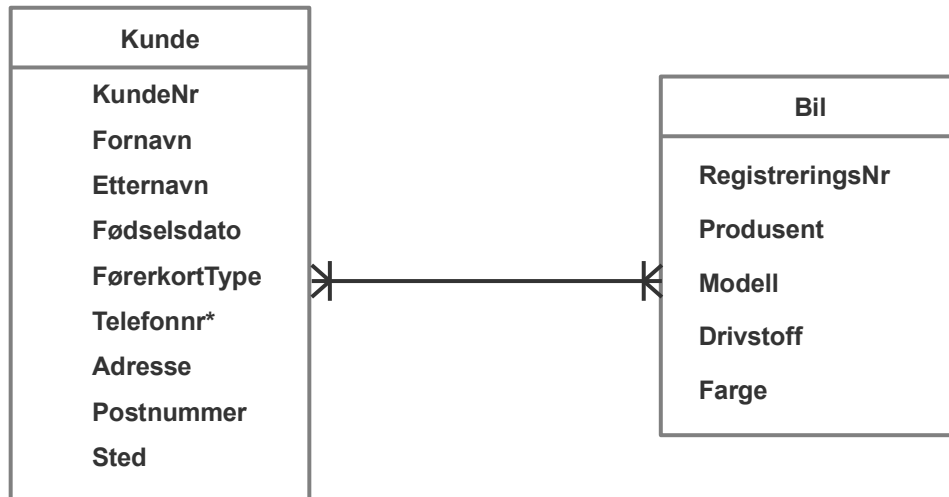
Mange-til-mange sammenhenger

- En mange-til-mange sammenheng viser et problem med designen av databasen
 - Kan ikke realiseres i en relasjonsdatabase
- Illustrerer områder med dårlig forståelse
- En mange-til-mange sammenheng skjuler en skjult entitet
- Behandles ved å identifisere og oppløse den opprinnelige *mange-til-mange* til to *én-til-mange* relasjoner



ER modellering praktisk

- Hvis vi går tilbake til vår bilutleie eksempel



*Telefonnr er en flerverdi attributt

59/171



Konvertering til logisk skjema

1. Entitetstyper blir relasjoner
2. Mange-til-mange gjøres om til egen relasjon
3. Flerverdi attributter gjøres om til egen relasjon
4. Fremmednøkler erstatter én-til-mange på mange siden
5. Fremmednøkler erstatter én-til-én sammenhenger

60/171



Entitetstyper blir relasjoner

Kunde (KundeNr, Fornavn, Etternavn, Fødselsdato, FørerkortType, TelefonNr, Adresse, Postnummer, Sted)

Bil (Registreringsnr, Produsent, Modell, Drivstoff, Farge)

61/171

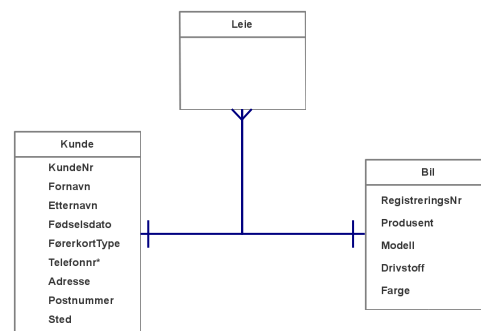


Mange-til-mange gjøres om til egen relasjon

Kunde (KundeNr, Fornavn, Etternavn, Fødselsdato, FørerkortType, Telefonnr, Adresse, Postnummer, Sted)

Bil (Registreringsnr, Produsent, Modell, Drivstoff, Farge)

Leie ()



62/171



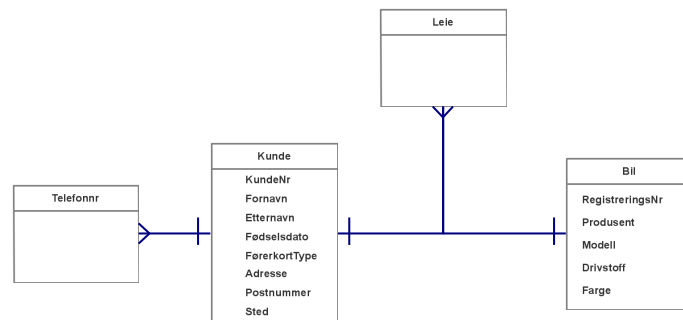
Flerverdi attributter gjøres om til egen relasjon

Kunde (KundeNr, Fornavn, Etternavn, Fødselsdato, FørerkortType, Adresse, Postnummer, Sted)

Bil (Registreringsnr, Produsent, Modell, Drivstoff, Farge)

Leie ()

Telefonnr ()



63/171



Fremmednøkler erstatter èn-til-mange på mange siden

Kunde (KundeNr*, Fornavn, Etternavn, Fødselsdato, FørerkortType, Adresse, Postnummer, Sted)

Bil (Registreringsnr*, Produsent, Modell, Drivstoff, Farge)

Leie (KundeNr, Registreringsnr, FraDato, TilDato, KmStand)

Telefonnr (KundeNr, Telefonnr)

64/171



SQL Introduksjonskurs

CREATE, ALTER, DROP,
TRUNCATE, INSERT,
DELETE, UPDATE



Datatyper

- I en database må datatypen til hvert eneste attributt (kolonne) spesifiseres
- De fleste DBMS støtter et sett forskjellige datatyper som kan brukes til å spesifisere en attributt
 - Datatypen gjelder alle verdier i attributtet
- Disse kan feks være:
 - Streng
 - Tall
 - Dato/Klokkeslett
- Vi ser på datatypene til MySQL

Datatyper (streng)

Datatype	Type	Beskrivelse
CHAR ()	Fast	Et felt av fast lengde mellom 0 og 255 tegn, der lengden kan spesifiseres
VARCHAR ()	Variabel	Et felt av variabel lengde mellom 0 og 65,535 tegn*, der lengden kan spesifiseres
TINYTEXT	Variabel	Et felt som kan bestå av maksimalt 255 tegn
TEXT	Variabel	En streng som kan bestå av maksimalt 65535 tegn
MEDIUMTEXT	Variabel	En streng som kan bestå av maksimalt 16777215 tegn
LONGTEXT	Variabel	En streng som kan bestå av maksimalt 4294967295 tegn

67/171

*Før MySQL 5.0.3 var VARCHAR mellom 0 og 255. Nå er det mellom 0 og 65,535 tegn



Datatyper (binærstreng)

- **Binary Large Object** er et felt som kan brukes til å lagre binær data feks en DOC file
 - Tenk på en DOC fil som en binær streng

Datatype	Beskrivelse
TINYBLOB	Felt med maksimal lengde på 255 tegn
BLOB	Felt med maksimal lengde på 65535 tegn
MEDIUMBLOB	Felt med maksimal lengde på 16777215 tegn
LARGEBLOB	Felt med maksimal lengde på 4294967295 tegn

68/171

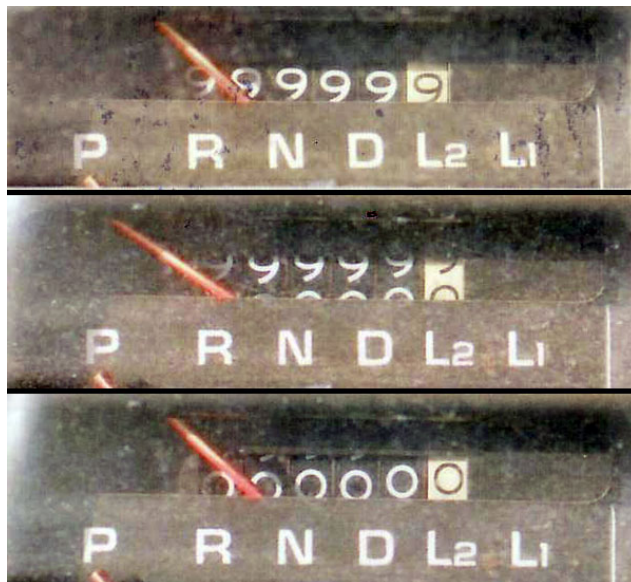


Datatyper (heltall)

- Beregninger (feks snitt på alder) blir utført fortere når et attributt spesifiseres som et tall
- Det er viktig å vite hva maks og min verdien av datatypen
 - Maksverdi + 1 = Minverdi

Datatype	Beskrivelse
TINYINT	Et tall mellom 0 og 255 eller -128 og 127
SMALLINT	Et tall mellom 0 og 65535 eller -32768 og 32767
MEDIUMINT	Et tall mellom 0 og 65535 eller -32768 og 32767
INT	Et tall mellom 0 og 16777215 eller -8388608 og 8388607
BIGINT	Et tall mellom 0 og 18446744073709551615 eller -9223372036854775808 og 9223372036854775807

Data typer (hellettall)



Datatyper (andre tall)

- Desimal tall feks 129.7777 bruker følgende data typer
 - **FLOAT**
 - **DOUBLE**
 - **DECIMAL**
- **BOOLEAN**
 - Ja eller Nei

Datatyper (dato og tid)

Datatype	Beskrivelse
DATE	En dato mellom 1000-01-01 til 9999-12-31
DATETIME	En dato og klokkeslett mellom 1000-01-01 00:00:00 to 9999-12-31 23:59:59
TIME	-838:59:59 til 838:59:59
YEAR[(2 4)]	1901 to 2155
TIMESTAMP	Antall sekunder siden et bestemt klokkeslett

Data Definition Language (DDL)

- DDL kommandoer
 - **CREATE**
 - Opprett
 - **DROP**
 - Slett
 - **ALTER**
 - Endre
 - **TRUNCATE**
 - Tøm

73/171



DDL CREATE

- **CREATE** kan brukes til å lage følgende:
 - Database
 - Tabell (Table)
 - Bruker (User)
 - Indeks (Index)*

**Indeks tar vi på dag 2*

74/171



CREATE Syntaks

```
CREATE  
{DATABASE | SCHEMA}  
[IF NOT EXISTS]  
db_navn
```

Symbol forklaring

- {}
 - Du må velge et element fra et sett av elementer
- |
 - enten eller
 - Feks **select** | **insert**
 - Det er enten **select** eller **insert**
- []
 - valgfri del av kommandoen

CREATE

```
CREATE DATABASE  
etternavn_noark5;
```

Sett inn etternavnet ditt her for å
unngå at alle samtidig prøver å lage
samme database.

77/171



USE database

- Du må vanligvis angi hvilken database du jobber mot

```
CREATE DATABASE etternavn_noark5;  
USE etternavn_noark5;
```

78/171



CREATE (TABLE)

```
CREATE  
TABLE [tabellnavn]  
(  
[kolonne egenskaper]  
)
```

Husk du må spesifisere gjeldende database før du kan lage en tabell

79/171



Så lager vi en tabell

```
CREATE TABLE  
arkiv  
(  
[kolonne egenskaper]  
);
```

80/171



Metadata for <arkiv>

Nr.	Navn	Forek.	Datatype
M001	systemID	1	Tekststreng
M020	tittel	1	Tekststreng
M021	beskrivelse	0-1	Tekststreng
M050	arkivstatus	0-1	Tekststreng
M300	dokumentmedium	0-1	Tekststreng
M301	oppbevaringssted	0-M	Tekststreng
M600	opprettetDato	1	Dato og klokkeslett
M601	opprettetAv	1	Tekststreng
M602	avsluttetDato	1	Dato og klokkeslett
M603	avsluttetAv	1	Tekststreng

DDL arkiv

```
CREATE TABLE arkiv (  
  systemId CHAR(36),  
  tittel VARCHAR(100) NOT NULL,  
  beskrivelse VARCHAR(100),  
  arkivstatus CHAR(20),  
  dokumentmedium VARCHAR(50),  
  opprettetDato DATETIME NOT NULL,  
  opprettetAv VARCHAR(100) NOT NULL,  
  avsluttetDato DATETIME,  
  avsluttetAv VARCHAR(100)  
) engine = InnoDB;
```

DDL arkiv med primærnøkkel

```
CREATE TABLE arkiv (  
  systemId CHAR(36) PRIMARY KEY,  
  tittel VARCHAR(100) NOT NULL,  
  beskrivelse VARCHAR(100),  
  arkivstatus CHAR(20),  
  dokumentmedium VARCHAR(50),  
  opprettetDato DATETIME NOT NULL,  
  opprettetAv VARCHAR(100) NOT NULL,  
  avsluttetDato DATETIME,  
  avsluttetAv VARCHAR(100)  
) engine = InnoDB;
```

83/171



DDL arkiv med primærnøkkel

```
CREATE TABLE arkiv (  
  systemId CHAR(36),  
  tittel VARCHAR(100) NOT NULL,  
  beskrivelse VARCHAR(100),  
  arkivstatus CHAR(20),  
  dokumentmedium VARCHAR(50),  
  opprettetDato DATETIME NOT NULL,  
  opprettetAv VARCHAR(100) NOT NULL,  
  avsluttetDato DATETIME,  
  avsluttetAv VARCHAR(100),  
  PRIMARY KEY (systemId)  
) engine = InnoDB;
```

84/171

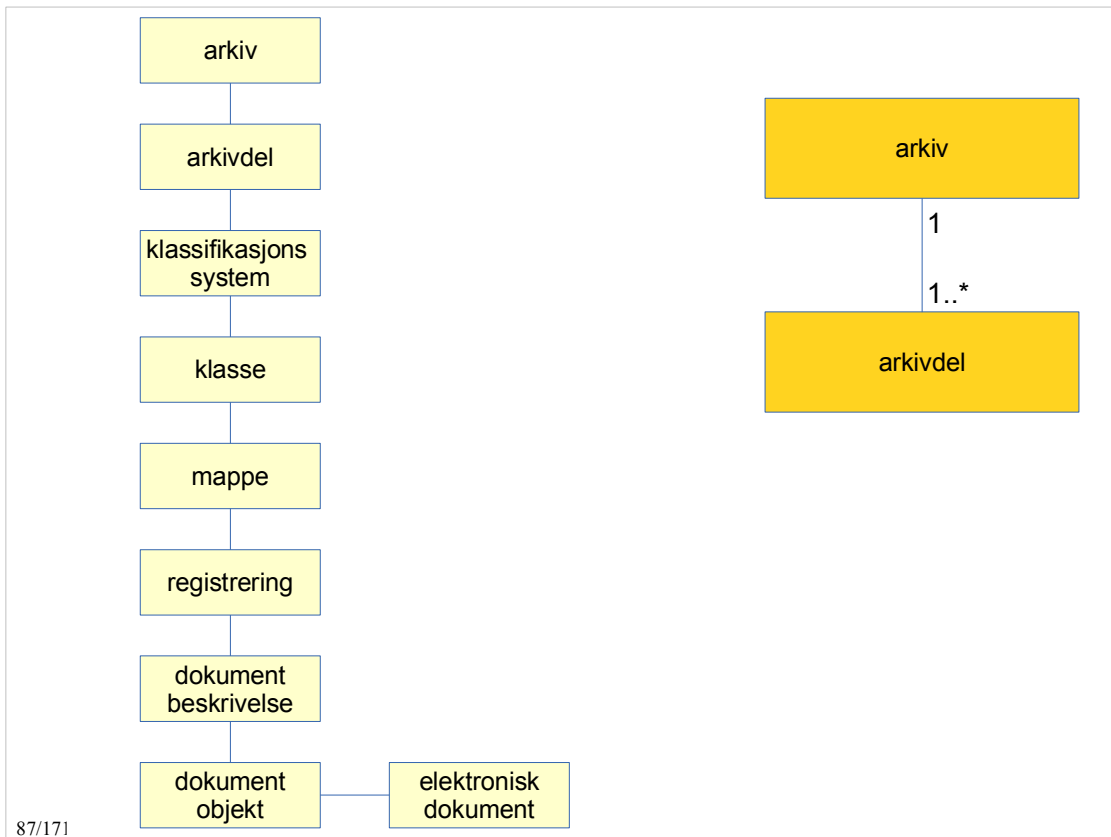


Metadata for <arkivdel>

Nr.	Navn	Forek	Datatype
M001	systemID	1	Tekststreng
M020	tittel	1	Tekststreng
M021	beskrivelse	0-1	Tekststreng
M051	arkivdelstatus	1	Tekststreng
M300	dokumentmedium	0-1	Tekststreng
M301	oppbevaringssted	0-M	Tekststreng
M600	opprettetDato	1	Dato og klokkeslett
M601	opprettetAv	1	Tekststreng
M602	avsluttetDato	1	Dato og klokkeslett
M603	avsluttetAv	1	Tekststreng
M107	arkivperiodeStartDato	0-1	Dato
M108	arkivperiodeSluttDato	0-1	Dato
M202	referanseForloeper	0-1	arkivdel.systemID
M203	referanseArvtaker	0-1	arkivdel.systemID

DDL arkivdel

```
CREATE TABLE arkivdel (  
  systemId CHAR(36),  
  tittel VARCHAR(100) NOT NULL,  
  beskrivelse VARCHAR(100),  
  arkivdelstatus CHAR(20) NOT NULL,  
  dokumentmedium VARCHAR(50),  
  opprettetDato DATETIME NOT NULL,  
  opprettetAv VARCHAR(100) NOT NULL,  
  avsluttetDato DATETIME,  
  avsluttetAv VARCHAR(100),  
  arkivperiodeStartDato DATE,  
  arkivperiodeSluttDato DATE,  
  referanseForloeper CHAR(36),  
  referanseArvtaker CHAR(36),  
  PRIMARY KEY (systemId)  
) engine = InnoDB;
```

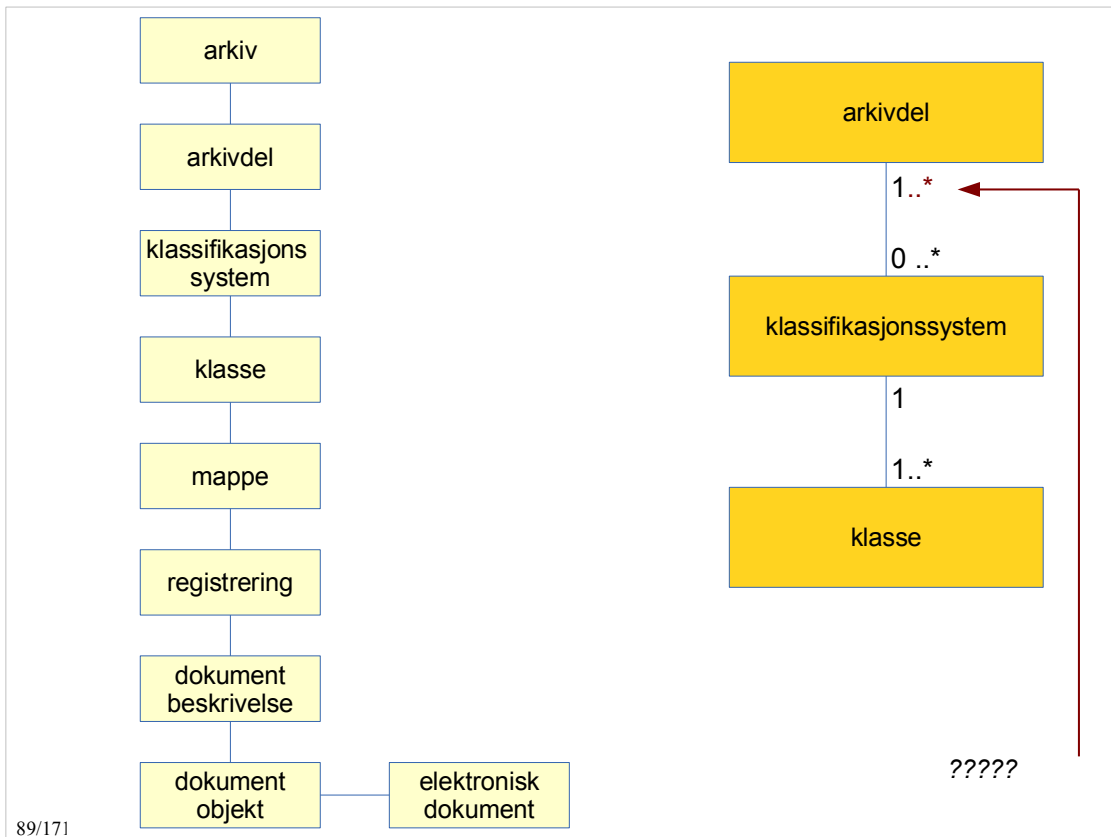


DDL arkivdel med fremmednøkkel

```

CREATE TABLE arkivdel (
  systemId CHAR(36),
  tittel VARCHAR(100) NOT NULL,
  beskrivelse VARCHAR(100),
  arkivdelstatus CHAR(20) NOT NULL,
  dokumentmedium VARCHAR(50),
  opprettetDato DATETIME NOT NULL,
  opprettetAv VARCHAR(100) NOT NULL,
  avsluttetDato DATETIME,
  avsluttetAv VARCHAR(100),
  arkivperiodeStartDato DATE,
  arkivperiodeSluttDato DATE,
  referanseForloeper CHAR(36),
  referanseArvtaker CHAR(36),
  referanseArkiv CHAR(36),
  PRIMARY KEY (systemId)
) engine = InnoDB;
  
```

Ikke en del av Noark standarden



DDL arkivdel med fremmednøkkel

```

CREATE TABLE arkivdel (
  systemId CHAR(36),
  tittel VARCHAR(100) NOT NULL,
  beskrivelse VARCHAR(100),
  arkivdelstatus CHAR(20) NOT NULL,
  dokumentmedium VARCHAR(50),
  opprettetDato DATETIME NOT NULL,
  opprettetAv VARCHAR(100) NOT NULL,
  avsluttetDato DATETIME,
  avsluttetAv VARCHAR(100),
  arkivperiodeStartDato DATE,
  arkivperiodeSluttDato DATE,
  referanseForloeper CHAR(36),
  referanseArvtaker CHAR(36),
  referanseArkiv CHAR(36),
  referanseKlassifikasjonssystem CHAR(36),
  FOREIGN KEY (referanseArkiv) REFERENCES arkiv (systemID),
  FOREIGN KEY (referanseKlassifikasjonssystem)
    REFERENCES klassifikasjonssystem (systemID),
  PRIMARY KEY (systemId)
) engine = InnoDB;

```

Metadata for klassifikasjonssystem

Nr.	Navn	Forek	Avl	Datatype
M001	systemID	1	A	Tekststreng
M086	klassifikasjonstype	0-1	A	Tekststreng
M020	tittel	1	A	Tekststreng
M021	beskrivelse	0-1	A	Tekststreng
M600	opprettetDato	1	A	Dato og klokkeslett
M601	opprettetAv	1	A	Tekststreng
M602	avsluttetDato	0-1	A	Dato og klokkeslett
M603	avsluttetAv	0-1	A	Tekststreng

DDL klassifikasjonssystem

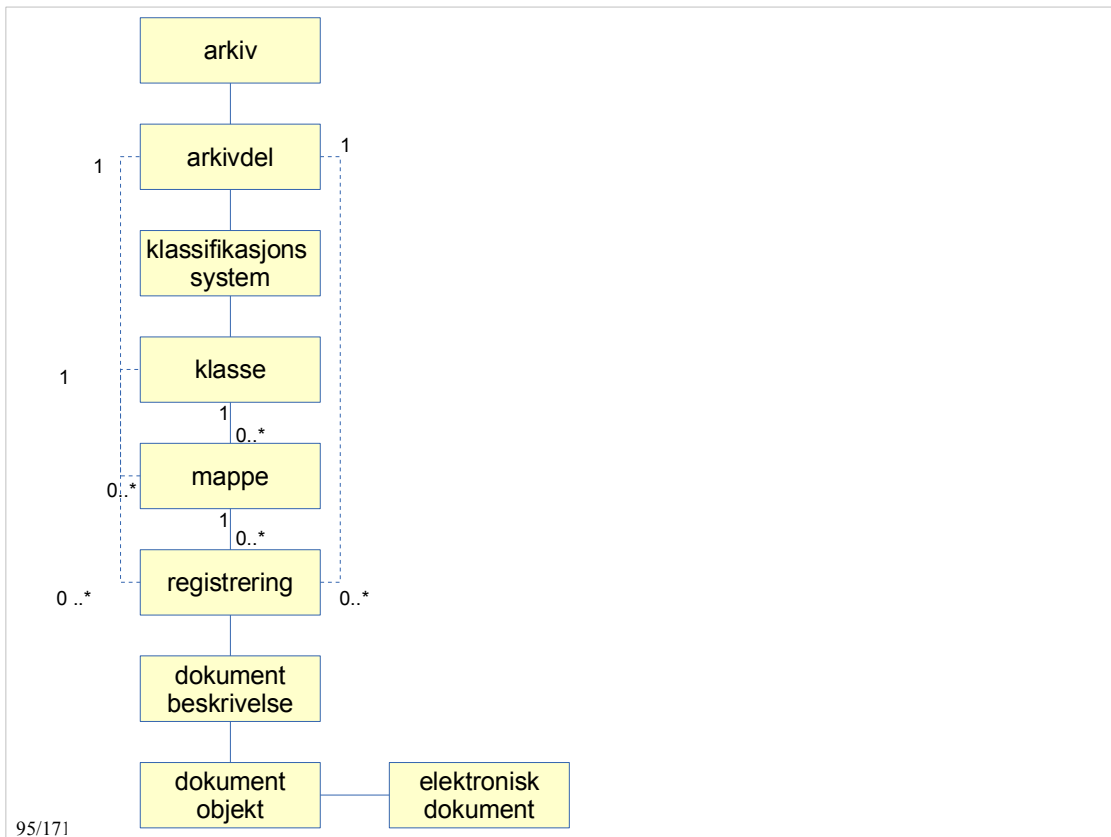
```
CREATE TABLE klassifikasjonssystem(  
  systemId CHAR(36),  
  klassifikasjonstype VARCHAR(100),  
  tittel VARCHAR(100),  
  beskrivelse VARCHAR(100),  
  opprettetDato DATETIME,  
  opprettetAv VARCHAR(100),  
  avsluttetDato DATETIME,  
  avsluttetAv VARCHAR(100),  
  PRIMARY KEY (systemId)  
) engine = InnoDB;
```

Metadata for klasse

Navn	Forek	Avl.	Datatype
systemId	1	A	Tekststreng
klasseld	1	A	Tekststreng
tittel	1	A	Tekststreng
beskrivelse	0-1	A	Tekststreng
opprettetDato	1	A	Dato og klokkeslett
opprettetAv	1	A	Tekststreng
avsluttetDato	0-1	A	Dato og klokkeslett
avsluttetAv	0-1	A	Tekststreng

DDL klasse

```
CREATE TABLE klasse (  
  systemId CHAR(36) NOT NULL,  
  klasseld CHAR(36) NOT NULL,  
  tittel VARCHAR(100),  
  beskrivelse VARCHAR(100),  
  opprettetDato DATETIME NOT NULL,  
  opprettetAv VARCHAR(100) NOT NULL,  
  avsluttetDato DATETIME,  
  avsluttetAv VARCHAR(100),  
  referanseKlassifasjonssystem CHAR(36),  
  FOREIGN KEY (referanseKlassifasjonssystem)  
    REFERENCES klassifasjonssystem (systemID),  
  PRIMARY KEY (systemId)  
) engine = InnoDB;
```



Metadata for mappe

Nr.	Navn	Forek	Datatype
M001	systemID	1	Tekststreng
M003	mappelD	1	Tekststreng
M020	tittel	1	Tekststreng
M025	offentligTittel	0-1	Tekststreng
M021	beskrivelse	0-1	Tekststreng
M022	noekkelord	0-M	Tekststreng
M300	dokumentmedium	0-1	Tekststreng
M301	oppbevaringssted	0-M	Tekststreng
M600	opprettetDato	1	Dato og klokkeslett
M601	opprettetAv	1	Tekststreng
M602	avsluttetDato	1	Dato og klokkeslett
M603	avsluttetAv	1	Tekststreng
M208	referanseArkivdel	0-M	arkivdel.systemID
M711	virksomhetsspesifikke Metadata	0-1	Vilkårlig struktur

DDL mappe

```
CREATE TABLE mappe (  
  systemId CHAR(36) NOT NULL,  
  mappeld CHAR(36) NOT NULL,  
  tittel VARCHAR(100),  
  offentligTittel VARCHAR(100),  
  beskrivelse VARCHAR(100),  
  dokumentmedium VARCHAR(50),  
  opprettetDato DATETIME NOT NULL,  
  opprettetAv VARCHAR(100) NOT NULL,  
  avsluttetDato DATETIME,  
  avsluttetAv VARCHAR(100),  
  referanseArkivdel CHAR(36),  
  referanseKlasse CHAR(36),  
  FOREIGN KEY (referanseKlasse)  
    REFERENCES klasse (systemID),  
  FOREIGN KEY (referanseArkivdel)  
    REFERENCES arkivdel (systemID),  
  PRIMARY KEY (systemId)  
) engine = InnoDB;
```

97/171



Metadata for registrering

Nr.	Navn	Forek	Avl	Datatype
M001	systemID	1	A	Tekststreng
M600	opprettetDato	1	A	Dato og klokkeslett
M601	opprettetAv	1	A	Tekststreng
M604	arkivertDato	1	A	Dato og klokkeslett
M605	arkivertAv	1	A	Tekststreng
M208	referanseArkivdel	0-M	A	arkivdel.systemID

98/171

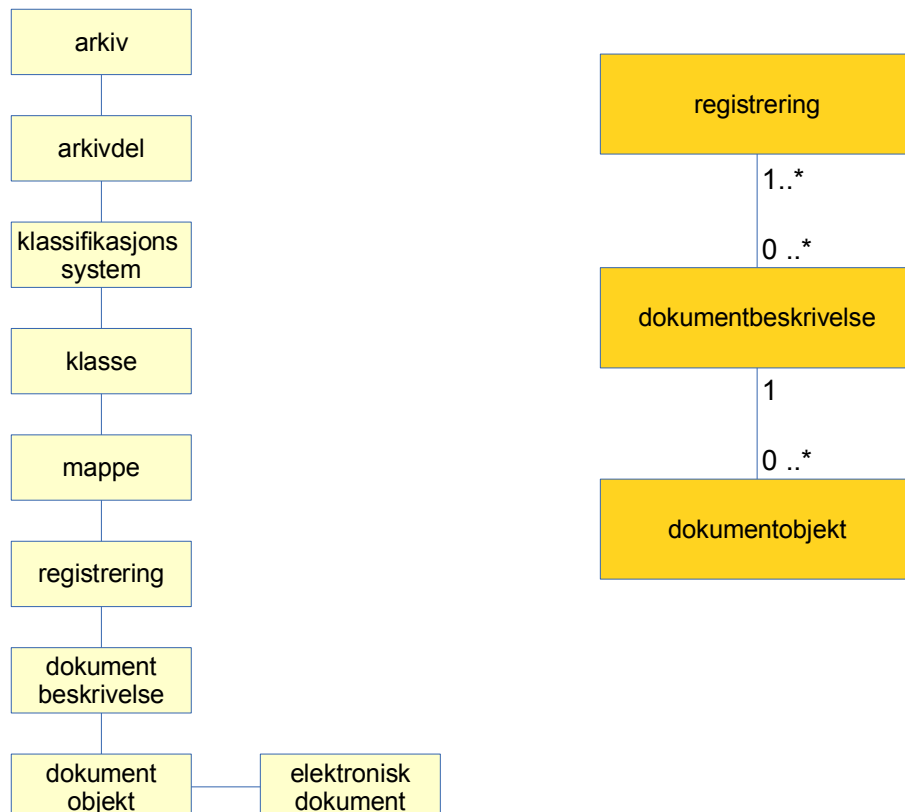


DDL registrering

```
CREATE TABLE registrering (  
  systemId CHAR(36),  
  opprettetDato DATETIME NOT NULL,  
  opprettetAv VARCHAR(100) NOT NULL,  
  arkivertDato DATETIME,  
  arkivertAv VARCHAR(100),  
  referanseArkivdel CHAR(36),  
  referanseMappe CHAR(36),  
  referanseKlasse CHAR(36),  
  FOREIGN KEY (referanseArkivdel) REFERENCES arkivdel (systemID),  
  FOREIGN KEY (referanseMappe) REFERENCES mappe (systemID),  
  FOREIGN KEY (referanseKlasse) REFERENCES klasse (systemID),  
  PRIMARY KEY (systemId)  
) engine = InnoDB;
```

#Dette er 0-1, ikke 0-M

99/171



100/17

Metadata for <dokumentbeskrivelse>

Nr.	Navn	Forek	Datatype
M001	systemID	1	Tekststreng
M083	dokumenttype	1	Tekststreng
M054	dokumentstatus	1	Tekststreng
M020	tittel	1	Tekststreng
M021	beskrivelse	0-1	Tekststreng
M024	forfatter	0-M	Tekststreng
M600	opprettetDato	1	Dato og klokkeslett
M601	opprettetAv	1	Tekststreng
M300	dokumentmedium	0-1	Tekststreng
M301	oppbevaringssted	0-1	Tekststreng
M208	referanseArkivdel	0-M	arkivdel.systemID
M217	tilknyttetRegistreringSom	1	Tekststreng
M007	dokumentnummer	1	Heltall
M620	tilknyttetDato	1	Dato og klokkeslett
M621	tilknyttetAv	1	Tekststreng

DDL dokumentbeskrivelse

```
CREATE TABLE dokumentbeskrivelse (  
  systemId CHAR(36),  
  dokumenttype VARCHAR(100) NOT NULL,  
  dokumentstatus VARCHAR(100) NOT NULL,  
  tittel VARCHAR(100),  
  beskrivelse VARCHAR(100),  
  forfatter VARCHAR(255),  
  opprettetDato DATETIME,  
  opprettetAv VARCHAR(100),  
  dokumentmedium VARCHAR(50),  
  referanseArkivdel CHAR(36),  
  tilknyttetRegistreringSom VARCHAR(100) NOT NULL,  
  tilknyttetDato DATETIME,  
  tilknyttetAv VARCHAR(100),  
  dokumentnummer INT,  
  referanseMappe CHAR(36),  
  referanseRegistrering CHAR(36), #Dette er 0-1, ikke 1-M  
  FOREIGN KEY (referanseMappe) REFERENCES mappe (systemID),  
  FOREIGN KEY (referanseRegistrering )  
    REFERENCES registrering (systemID),  
  PRIMARY KEY (systemId)  
) engine = InnoDB;
```

Metadata for <dokumentobjekt>

Nr.	Navn	Forek.	Datatype
M005	versjonsnummer	1	Heltall
M700	variantformat	1	Tekststreng
M701	format	1	Tekststreng
M702	formatDetaljer	0-1	Tekststreng
M600	opprettetDato	1	Dato og klokkeslett
M601	opprettetAv	1	Tekststreng
M218	referanseDokumentfil	1	Tekststreng (filkatalogstruktur + filnavn)
M705	sjekksum	1	Tekststreng
M706	sjekksumAlgoritme	1	Tekststreng
M707	filstoerrelse	1	Tekststreng

DDL dokumentobjekt

```
CREATE TABLE dokumentobjekt(  
  systemId CHAR(36),  
  versjonsnummer INT NOT NULL,  
  variantformat VARCHAR(255) NOT NULL,  
  format VARCHAR(255) NOT NULL,  
  formatDetaljer VARCHAR(255) ,  
  opprettetDato DATETIME NOT NULL,  
  opprettetAv VARCHAR(255) NOT NULL,  
  referanseDokumentfil VARCHAR(255) NOT NULL,  
  sjekksum VARCHAR(255) NOT NULL,  
  sjekksumAlgoritme VARCHAR(255) NOT NULL,  
  filstoerrelse VARCHAR(255) NOT NULL,  
  dokumentbeskrivelse VARCHAR(255) NOT NULL,  
  referanseDokumentbeskrivelse CHAR(36),  
  PRIMARY KEY (systemId),  
  FOREIGN KEY (referanseDokumentbeskrivelse)  
    REFERENCES dokumentbeskrivelse (systemID)  
) engine = InnoDB;
```

CREATE

```
CREATE USER bruker  
[IDENTIFIED BY  
PASSWORD] 'password'
```

DROP Syntaks

```
DROP  
{DATABASE | SCHEMA}  
[IF EXISTS]  
db_navn
```

DROP

DROP {DATABASE | SCHEMA} [IF EXISTS] db_navn;

DROP TABLE [IF EXISTS]
tbl_navn [, tbl_navnN] ...

DROP USER bruker [, brukerN] ...

DROP INDEX indeks_navn ON tbl_navn

ALTER

- Etter du har opprettet en tabell, kan du ha behov for å endre det, feks
 - Legge til en ny (kolonne | indeks | begrensning) til en tabell
 - Slette en (kolonne | indeks | begrensning) fra en tabell
 - Endre primærnøkkelen

ALTER

```
ALTER TABLE dokumentobjekt ADD FOREIGN KEY  
  (referanseDokumentbeskrivelse)  
  REFERENCES dokumentbeskrivelse (systemId);
```

```
ALTER TABLE dokumentobjekt DROP dokumentbeskrivelse;
```

```
ALTER TABLE dokumentobjekt ADD dokumentbeskrivelse  
  VARCHAR(255) NOT NULL;
```

DML INSERT

- Data settes inn i en relasjon en tuppel om gangen
- Det minste du trenger å ha med er primærnøkkelen til tuppleen
 - Hvis det ikke er noen begrensninger på andre attributter feks NOT NULL
- Hvis du har mange rader av data som skal settes inn så gjøres det en tuppel om gangen
- Data som går over 2 tuppler i forskjellige relasjoner kan settes inn samtidig
 - Det er ingen krav om transaksjon, men dumt å ikke bruke det
 - Husk referanseintegriteten

Data Manipulation Language (DML)

- DML kommandoer
 - **INSERT**
 - Sett inn
 - **UPDATE**
 - Oppdater
 - **DELETE**
 - Slett

**INSERT INTO TABELL
VALUES ();**

INSERT INTO TABELL (attributtliste) VALUES (verdiliste);

INSERT arkiv

```
INSERT INTO arkiv (  
  systemId, tittel,  
  beskrivelse, arkivstatus, dokumentmedium,  
  opprettetDato, opprettetAv)  
VALUES  
(  
  "b1b6b3f1-9186-4899-b3d9-4ebf0174727c", "KDRS Arkivet",  
  "Test arkiv for kurs", "Opprettet", "Elektronisk arkiv",  
  "2014-03-13", "admin"  
);
```

INSERT arkivdel

```
INSERT INTO arkivdel (  
  systemId, tittel, beskrivelse,  
  arkivdelstatus, dokumentmedium, opprettetDato, opprettetAv,  
  arkivperiodeStartDato, referanseArkiv  
)  
VALUES  
(  
  "6cf257fc-262b-4a40-9e00-766d718d8a64", "Arkivdel 1", null,  
  "Aktiv periode", "Elektronisk arkiv", "2014-03-13", "admin",  
  "2014-03-13", "b1b6b3f1-9186-4899-b3d9-4ebf0174727c"  
);
```

UPDATE SET WHERE;

*UPDATE TABELL
SET
Attributt='verdi'
WHERE;*

UPDATE arkivdel

```
UPDATE arkivdel SET  
  beskrivelse = "Oppdatert beskrivelse"  
WHERE  
  systemId = "6cf257fc-262b-4a40-9e00-766d718d8a64";
```

```
UPDATE arkivdel SET  
  arkivdelstatus = "Avsluttet periode"  
WHERE  
  arkivdelstatus = "Aktiv periode";
```

DELETE
FROM
WHERE;

*DELETE
FROM TABELL
WHERE;*

DELETE arkivdel

Fungerer dette?

```
DELETE FROM arkiv WHERE
```

```
  systemId = "b1b6b3f1-9186-4899-b3d9-4ebf0174727c";
```

Hva med dette?

```
DELETE FROM arkivdel WHERE
```

```
  systemId = "6cf257fc-262b-4a40-9e00-766d718d8a64";
```

```
DELETE FROM arkiv WHERE
```

```
  systemId = "b1b6b3f1-9186-4899-b3d9-4ebf0174727c";
```

Databaser og brukere

- Alle brukere er like, men noen brukere er mer like enn andre!
- Ikke alle brukere skal ha lik tilgang til utføre kommandoer
 - Det er mulig å begrense hvem som kan gjøre hva
- Data Control Language er brukt til denne oppgaven

Data Control Language (DCL)

- Data Control Language er et sett med kommandoer som brukes til
 - å gi rettigheter til noen med
 - **GRANT** kommandoen
 - å fjerne rettigheter fra noen med
 - **REVOKE** kommandoen
- Vi ser bare på noen få DCL kommandoer som gjelder for MySQL
 - Dere kan leke med dette hvis dere installerer MySQL server på maskinene deres

Syntaks (GRANT)

GRANT	GI
ON	PÅ
TO	TIL

Syntaks (GRANT)

```
GRANT PRIVILEGE  
ON OBJECT  
TO USER | GROUP;
```

Syntaks (GRANT)

```
GRANT { SELECT | INSERT |  
        UPDATE | DELETE |  
        REFERENCES |  
        TRIGGER }  
  
ON { tablename | functionname |  
    procedurename }  
  
TO { USER | GROUP }
```

Eksempel GRANT CREATE

- Gi en bruker rett til å lage tabeller i en database
- Gi en bruker rett til å lage en database med navn *dbname*

```
GRANT CREATE  
ON dbname.*  
TO tsodring;
```

Hvor:

- *noark5* er en database (alle tabeller)
- *tsodring* er en bruker i databasen

Eksempel GRANT SELECT

- Gi en bruker rett til å utføre spørringer på en database

```
GRANT SELECT  
ON noark5.*  
TO tsodring;
```

Hvor:

- *noark5* er en database (alle tabeller)
- *tsodring* er en bruker i databasen

Eksempel GRANT SELECT

- Gi en bruker rett til å utføre spørringer på en en bestemt tabell i en database

```
GRANT SELECT  
ON noark5.arkiv  
TO tsodring;
```

Hvor:

- *noark5* er en database
- *arkiv* er en tabell i databasen *noark5*
- *tsodring* er en bruker i databasen

Eksempel GRANT UPDATE

- Gi en bruker rett til å oppdatere en bestemt tabell i en database

```
GRANT UPDATE  
ON noark5.arkiv  
TO tsodring;
```

Hvor:

- *noark5* er en database
- *arkiv* er en tabell i databasen *noark5*
- *tsodring* er en bruker i databasen

Alle* Priviliger Som Kan Gies

Privilege	Meaning
ALL PRIVILEGES	Grant all privileges at specified access level except GRANT OPTION
ALTER	Enable use of ALTER TABLE
ALTER ROUTINE	Enable stored routines to be altered or dropped
CREATE	Enable database and table creation
CREATE ROUTINE	Enable stored routine creation
CREATE TEMPORARY TABLES	Enable use of CREATE TEMPORARY TABLE
CREATE USER	Enable use of CREATE USER , DROP USER , RENAME USER , and REVOKE ALL PRIVILEGES
CREATE VIEW	Enable views to be created or altered
DELETE	Enable use of DELETE
DROP	Enable databases, tables, and views to be dropped
EXECUTE	Enable the user to execute stored routines
FILE	Enable the user to cause the server to read or write files
GRANT OPTION	Enable privileges to be granted to or removed from other accounts
INDEX	Enable indexes to be created or dropped
INSERT	Enable use of INSERT
LOCK TABLES	Enable use of LOCK TABLES on tables for which you have the SELECT privilege
PROCESS	Enable the user to see all processes with SHOW PROCESSLIST
REFERENCES	Not implemented
RELOAD	Enable use of FLUSH operations
REPLICATION CLIENT	Enable the user to ask where master or slave servers are
REPLICATION SLAVE	Enable replication slaves to read binary log events from the master
SELECT	Enable use of SELECT
SHOW DATABASES	Enable SHOW DATABASES to show all databases
SHOW VIEW	Enable use of SHOW CREATE VIEW
SHUTDOWN	Enable use of mysqladmin shutdown
SUPER	Enable use of other administrative operations such as CHANGE MASTER TO , KILL , PURGE BINARY LOGS , SET GLOBAL , and mysqladmin debug command
UPDATE	Enable use of UPDATE
USAGE	Synonym for "no privileges"

*gjelder MySQL

*Det administratoren gir
kan administratoren ta*

Syntaks (REVOKE)

REVOKE	TA
ON	PÅ
FROM	FRA

Eksempel REVOKE SELECT

- Fjern en brukers rett til å utføre spørringer på en en bestemt tabell i en database

```
REVOKE SELECT  
ON noark5.arkiv  
FROM tsodring;
```

Hvor:

- **noark5** er en database
- **arkiv** er en tabell i databasen **noark5**
- **tsodring** er en bruker i databasen

Eksempel REVOKE SELECT

- Fjern en brukers rett til å oppdatere en bestemt tabell i en database

```
REVOKE UPDATE  
ON noark5.arkiv  
FROM tsodring;
```

Hvor:

- **noark5** er en database
- **arkiv** er en tabell i databasen **noark5**
- **tsodring** er en bruker i databasen

SQL Introduksjonskurs

SQL, SQL Aggregering, SQL Joins,
Indekser, Views,
Eksport og Import



Structured Query Language (SQL)

- SQL er en database språk utviklet for håndtering av data i relasjonsdatabaser
- Strengt tatt så inkluderer SQL DML, DDL og DCL kommandoer men for enkelthetskyld så bruker vi SQL kun som et spørrespråk

SELECT * FROM Bil;

RegistreringsNr	UnderstellsNr	Farge	Produsent	Modell
LH12984	10946534	Red	Volkswagen	Golf
DK23491	9648573	Blue	Toyota	Yaris
BP12349	5523840	Green	Skoda	Fabia
ZT97495	2643923	White	Seat	Leon

**SELECT Attributt FROM
Relasjon;**

SQL Eksempler

```
SELECT tittel FROM mappe;
```

```
SELECT mappeld, tittel FROM mappe;
```

```
SELECT * FROM mappe;
```

```
SELECT opprettetDato FROM registrering;
```

```
SELECT * FROM registrering;
```

De forrige eksemplene gir oss alle radene i tabellen men det er ofte du er ute etter en bestemt rad eller felt i databasen

For å filtrere bruker vi
WHERE

SELECT **Attributt(er)**
FROM **Relasjon**
WHERE **Attributt = 'verdi';**

SELECT med WHERE

```
SELECT * FROM mappe WHERE mappelid = '1';
```

```
SELECT * FROM arkivdel WHERE  
    arkivdelstatus = 'Aktiv periode';
```

```
SELECT * FROM registrering WHERE  
    opprettetDato = '2014-03-13';
```

```
SELECT * FROM dokumentobjekt WHERE  
    format = 'doc';
```

SQL og jokertegn

- SQL støtter *jokertegn* som kan brukes for å få flere søkeresultater
 - SQL bruker **%** og **_** som jokertegn
- Dette kan brukes til å finne tupler med lignende tekster
 - Kan være nyttig hvis du bare husker en del av en streng
- **'%'** og **'_'** tilsvarer **'*'** og **'?'** når du søker i operativsystemer

Eksempler på jokertegn

- Du søker på alle tittel i en sak som begynner med Oversikt
 - **LIKE 'Oversikt%'**
- Du søker på alle registrerings nummer der tall delen begynner med '12'
 - **LIKE '%12%'**
- Du søker på alle registrerings nummer som begynner med 'L' og tall delen er '12345'
 - **LIKE 'L_12345'**

SELECT med jokertegn

```
SELECT * FROM Bil WHERE  
  RegistreringsNr like 'LH%';
```

```
SELECT * FROM Bil WHERE  
  RegistreringsNr like '%12%';
```

```
SELECT * FROM Bil WHERE  
  RegistreringsNr like 'L_12345';
```

SQL DISTINCT

- Det kan hende at du ønsker en liste over bare de forskjellige eller distinkte verdiene i en attributt / tabell
 - Det er veldig nyttig å kunne identifisere de forskjellige filtypene i basen
 - doc, docx, xls, pdf osv
 - Det er også nyttig å kunne se hvilken status verdier er blitt brukt
 - Liste over forskjellige dokumentstatus

SELECT med DISTINCT

- Lag en liste over filformater i databasen

```
SELECT DISTINCT format FROM  
dokumentobjekt;
```

- Lag en liste over dokumentstatus i databasen

```
SELECT DISTINCT dokumentstatus FROM  
dokumentbeskrivelse;
```

SELECT med ORDER BY

- DBHS returnerer resultater i samme rekkefølge som den henter data ut av tabellen
- Noen ganger er det ønskelig å kunne sortere disse resultatene

```
SELECT Attributt(er)
FROM Relasjon
WHERE Attributt = 'verdi'
ORDER BY Attributt
[ ASC | DESC ] ;
```

SELECT med Order By

```
SELECT * FROM mappe
WHERE opprettetDato
  BETWEEN '2013-01-01 00:00:00' AND '2013-12-31 23:59:00';

SELECT * FROM mappe
WHERE opprettetDato
  BETWEEN '2013-01-01 00:00:00' AND '2013-12-31 23:59:00'
ORDER BY opprettetDato ASC;

SELECT * FROM mappe
WHERE opprettetDato
  BETWEEN '2013-01-01 00:00:00' AND '2013-12-31 23:59:00'
ORDER BY opprettetAv DESC;
```

SELECT og LIMIT

- Du kan begrense resultatsettet og spesifisere det maksimale antallet rader du vil ha tilbake

```
SELECT * FROM mappe LIMIT 3
```

- Du kan også feks begrense resultatsett ved å ignorere de første to og ta med de neste 5

```
SELECT * FROM mappe LIMIT 2,5
```

SQL GROUP BY

- En DBMS støtter en del innebygget beregningsfunksjoner
- Disse er brukt sammen med GROUP BY klausulen
 - AVG()
 - MAX()
 - MIN()
 - COUNT()
- MySQL støtter langt flere enn disse
 - Statistikk

SQL GROUP BY

- Dette er en veldig nyttig del av en DBMS og kan svare spørsmål som:
 - Hvor mange biler er det for hver farge?

```
SELECT Farge, COUNT( * ) AS TOTAL FROM Bil  
GROUP BY Farge;
```

- Hvor mange biler av typen 'Volkswagen Golf' er det for hver farge?

```
SELECT Farge, COUNT( * ) AS 'Golf farger' FROM  
Bil WHERE Modell = 'Golf' GROUP BY Farge;
```

SQL og JOINS

- Så langt har vi sett på tabeller der all data er i en enkel tabell
 - Dette tilsvarer jo flatefiler og viser ikke styrken i relasjonsmodellen som lar deg utføre spørringer som går på tvers av tabeller
- Data i tabeller kan kobles sammen på forskjellige måter
 - Natural Join
 - Anti Join
 - Right Outer Join
 - Left Outer Join
- Det er flere men vi holder oss til disse

SQL Natural Join

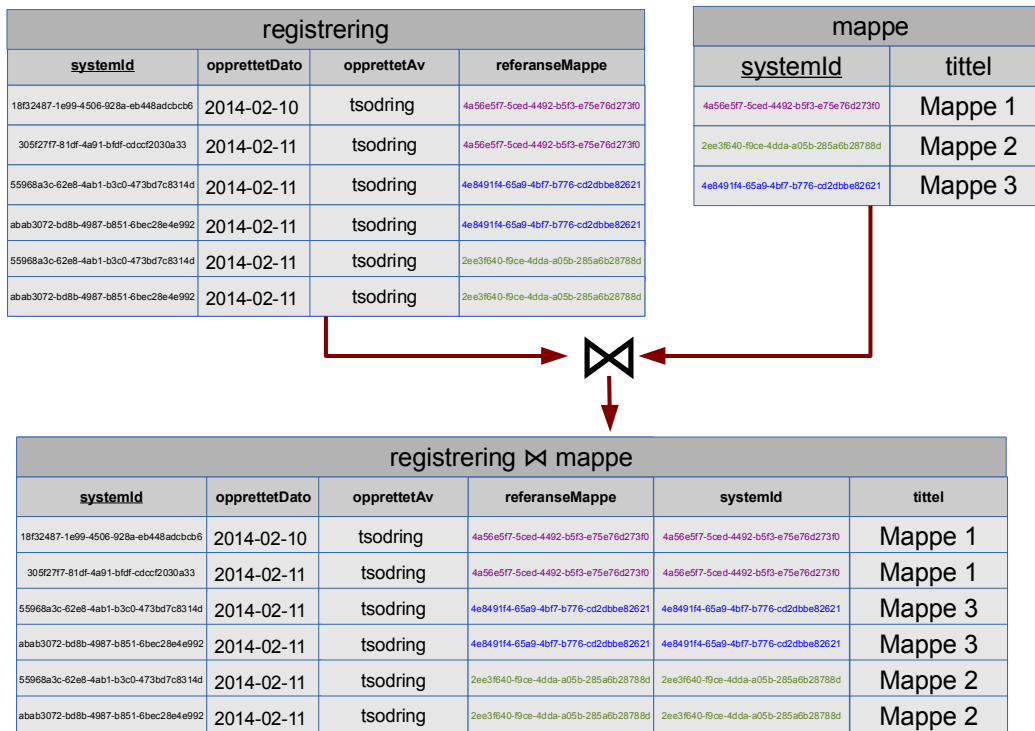
- En SQL Natural join mellom to tabeller A og B, er settet av alle kombinasjoner av tupler i A som har matchende tupler i B på felles attributten
- Lag en liste over registreringer og mappe informasjon

```
SELECT * FROM registrering, mappe
WHERE registrering.referanseMappe = mappe.systemId;
```

eller

```
SELECT * FROM registrering JOIN mappe ON
registrering.referanseMappe = mappe.systemId;
```

SQL Natural Join



SQL Antijoin

- En SQL *Antijoin* mellom to tabeller A og B, er settet av alle kombinasjoner av tupler i A som ikke har matchende tupler i B på felles attributtet
- Denne typen join kan svare spørsmålet
 - Hvilken registreringer har ingen mappe

```
SELECT * FROM registrering
WHERE (referanseMappe) NOT IN (
    SELECT systemId FROM mappe);
```

Men først

- Men med referanseintegritet så kan ikke denne situasjonen oppstå
- Før vi går videre så er vi nødt til å slette en tuppel fra *mappe* tabellen
 - `DELETE FROM mappe WHERE systemId = '4a56e5f7-5ced-4492-b5f3-e75e76d273f0';`
 - Dette er selvfølgelig imot referanseintegritet
- Vi må først fjerne begrensningen som ikke lar oss slette tuppelen
 - `ALTER TABLE mappe DROP FOREIGN KEY mappe_ibfk_1;`

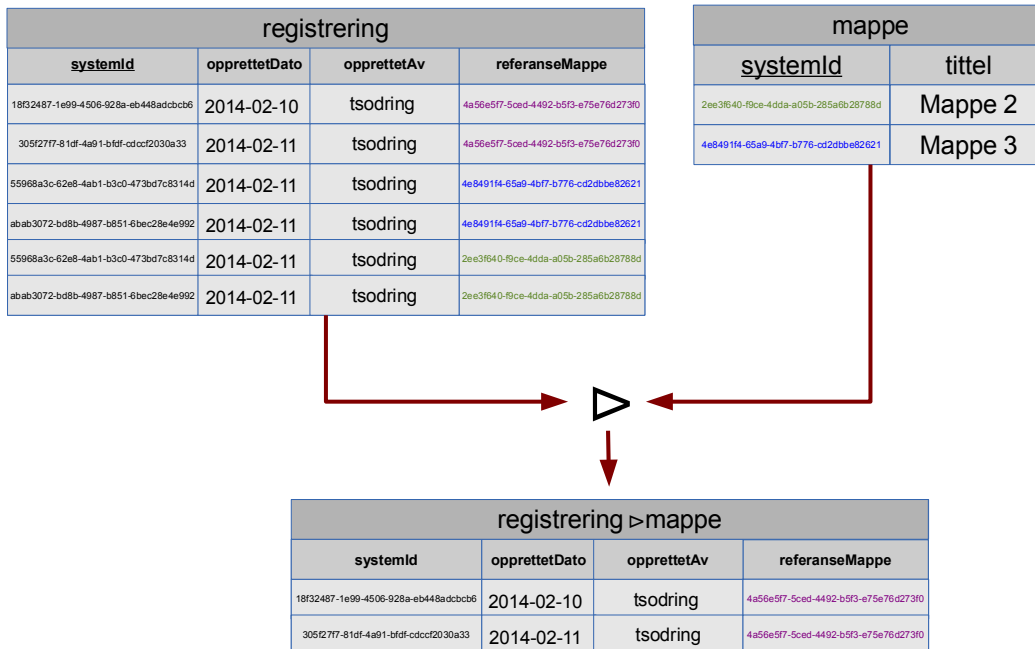
Fjern Begrensing

- Hvis **ALTER TABLE** kommandoen ikke fungerte
 - **SHOW CREATE TABLE** **mappe**;
- Finn teksten med følgende
 - **CONSTRAINT _____ FOREIGN KEY**
 - Feks **CONSTRAINT** **mappe_ibfk_1** **FOREIGN KEY**
- Utfør kommandoen:
 - **ALTER TABLE** **mappe** **DROP FOREIGN KEY** **mappe_ibfk_1**;  Disse skal være like!!
- Nå kan du utføre:
 - **DELETE FROM** **mappe** **WHERE** **systemId =** **'4a56e5f7-5ced-4492-b5f3-e75e76d273f0'**;

159/171



SQL AntiJoin

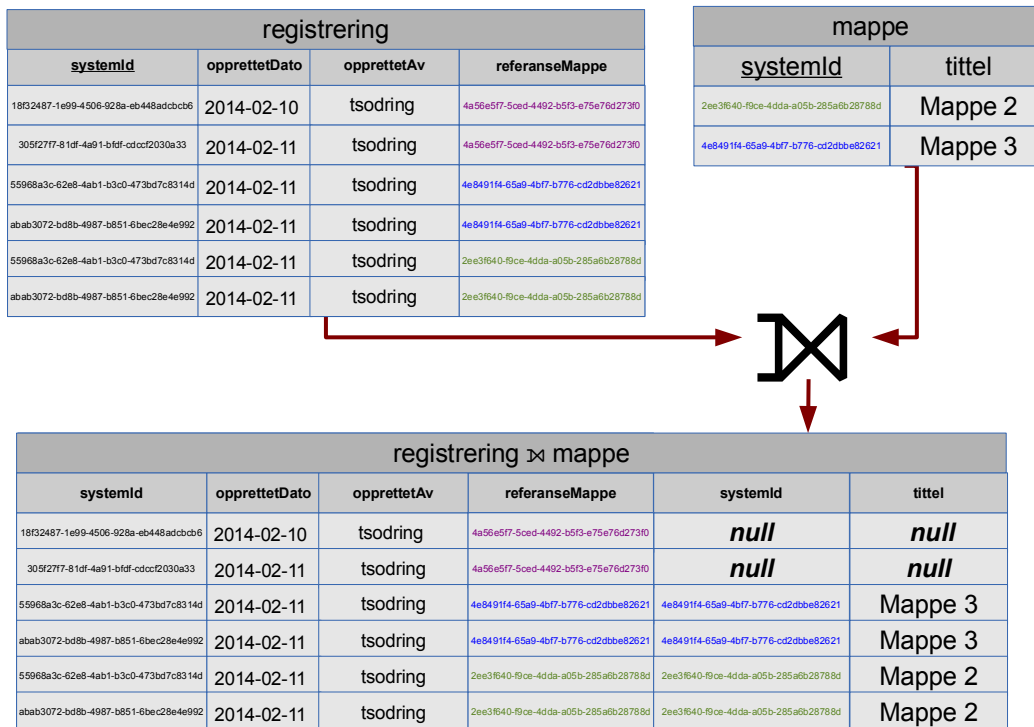


SQL Left Outer Join

- En SQL *Left Outer Join* mellom to tabeller, A og B, er settet av alle kombinasjoner av tupler i begge tabeller (A og B) som er lik på felles attributtet og de tuplene i A som ikke har matchende tupler i B

```
SELECT * FROM registrering
LEFT OUTER JOIN
mappe ON
registrering.referanseMappe = mappe.systemId;
```

SQL Left Outer Join

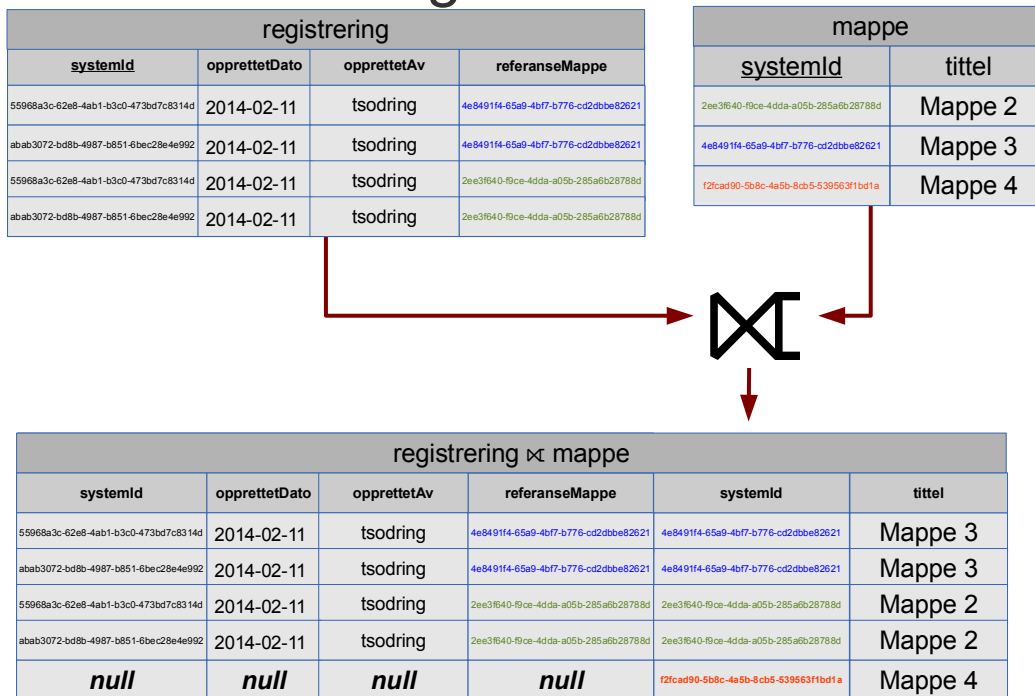


SQL Right Outer Join

- En SQL *Right Outer Join* mellom to tabeller, A og B, er settet av alle kombinasjoner av tupler i begge tabeller (A og B) som er lik på deres felles attributtet og de tuplene i B som ikke har tilsvarende tupler i A

```
SELECT * FROM registrering
RIGHT OUTER JOIN
mappe ON
registrering.referanseMappe = mappe.systemId;
```

SQL Right Outer Join



Indeks

- Hvis du ønsker å søke i en attributt (kolonne) med store mengder data så kan søket ta betydelig tid (mange sekunder)
 - Det finnes 2 900 000 biler i Norge, og hvis du søker på **UnderstellsNr** så kan det måtte søkes gjennom hele kolonnen før du finner
 - Hvis den du letter etter var nr 2 899 999 i kolonnen
 - DBHS støtter indekser som gjør at det er betydelig fortere å finne en bestemt tuppel
 - Forskjellen er alt fra minutter/sekunder til under en sekund avhengig av database størrelse
 - Primærnøkkelen er indeksert automatisk
 - Hver indeks øker størrelsen av databasen på disk

165/171



ALTER TABLE CREATE INDEX

```
ALTER TABLE Bil
ADD INDEX
    chassis_indeks (UnderstellsNr );
```

eller

```
ALTER TABLE Bil
ADD UNIQUE
    chassis_indeks (UnderstellsNr );
```

166/171



Views

- En view er en lagret SQL spørring som ser ut som en tabell og blir en slags virtuell tabell
- En view kan brukes til å
 - definere en undergruppe av tabeller / data
 - koble data fra flere tabeller (JOIN)
- Brukere kan få tilgangen begrenset til kun en view
- View kan spesifiseres med lese eller skrive tilgang på dataene

View med lesetilgang

- View som gir gjestetilgang til alle mapper, men brukeren kan kun se offentligTittel og opprettetDato

```
CREATE VIEW mappeOffentlig AS
SELECT mappeld, offentligTittel, opprettetDato
FROM mappe;

GRANT SELECT ON noark5.mappeOffentlig TO
gjest;
```

View for offentligjournal

- En view som utgjør offentligjournal

```
CREATE VIEW offentligJournal AS

SELECT jpst.journaldato, jpst.registreringsId,
       jpst.offentligTittel, jpst.beskrivelse,
       jpst.dokumentetsDato, saksmappe.saksansvarlig
FROM journalpost AS jpst
INNER JOIN saksmappe
ON referanseSaksmappe = saksmappe.systemId;
```

Eksport og Import (bare mysql)

```
# Maskin A
SELECT * FROM mappe
  INTO OUTFILE '/home/maskinAbruker/mappe.out'
  FIELDS TERMINATED BY ','
  ENCLOSED BY '"'
  LINES TERMINATED BY '\n';

# Maskin B
LOAD DATA LOCAL INFILE '/home/maskinBbruker/mappe.out'
  INTO TABLE mappe
  FIELDS TERMINATED BY ','
  ENCLOSED BY '"'
  LINES TERMINATED BY '\n';
```

Eksport og Import(kommandolinje)

- En fil bestående av sql kommandoer kan importeres i MySQL via terminalen
- De fleste DBHS støtter dette

```
(Maskin A) mysqldump -ubruker -pPassord -vdb_navn > db_navn_dump.sql
```

```
(Maskin B) mysqladmin -ubruker -pPassord create db_navn
```

```
(Maskin B) mysql -ubruker -pPassord db_navn < db_navn_dump.sql
```