



Økt 6 (av 12)

DB1100 Databaser

(Tomas Sandnes / tomas.sandnes@kristiania.no)

Dagens temaer

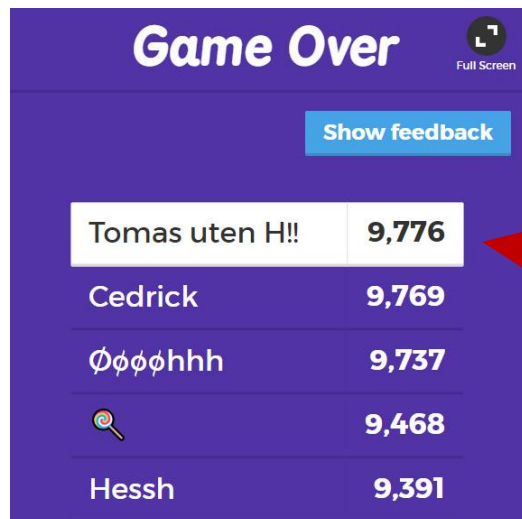
Dagens tema: [Relasjonsmodellen](#).

– Dagens pensum: [Læreboka, kapittel 6](#).

- Relasjonsalgebra
- Sniktitt på modellering
- Repetisjon
- (I øvingstimene etterpå: 2t multiple choice prøve, for egentesting.)

Resultat, forrige Kahoot

- Forrige gang var formen tilbake: Dere slo fjorårets studenter igjen! :-P
 - Stillingen er nå 3-1 i favør dere.
- Dere: $\approx 64\%$ riktig svar!
 - De: $\approx 59\%$ riktig svar.
 - Gi dere selv noen fortjente klapp på skulderen! :-D



Game Over	
Show feedback	
Tomas uten H!!	9,776
Cedrick	9,769
Øøøøhhh	9,737
🔍	9,468
Hessh	9,391

DB1100 - økt 5	
Played on	27 Sep 2018
Hosted by	Tomas_Sandnes
Played with	122 players
Played	10 of 10 questions

Overall Performance	
Total correct answers (%)	63,77%
Total incorrect answers (%)	36,23%
Average score (points)	5066,54 points

[Play Again](#)

[New Game](#)

"Tomas uten H!!" ???

- Haha, må innrømme at jeg har smilt av denne i ettertid.

- "Smooth", hvem det nå enn var. B-)

Lærdom fra Kahoot

Spm.4: Hvilken forutsetning er korrekt for oppdatering av underliggende data via views?

- A) Viewet kan referere til flere tabeller. (42 svar)
- B) DISTINCT kan være en del av viewet. (8 svar)
- C) GROUP BY eller HAVING kan benyttes i viewet. (23 svar)
- D) Alle elementer i view'ets select-del må være kolonner. (25 svar)

Kanskje ikke "... **oppdatering** av ..." kom godt nok frem i spm.?

- Vi må kunne finne frem til unike kolonner/celler i de underliggende tabellene for å kunne kjøre oppdateringer gjennom Views.
- Se kap 5.2.4 i pensumboka og slide 18 fra økt 5 (forrige gang).

Lærdom fra Kahoot #2

Sp.m9: Spørringen gir hva slags resultat?

```
SELECT * FROM city WHERE ID IN(SELECT capital FROM country);
```

A) Informasjon om alle hovedsteder. (24 svar)

B) Informasjon om alle byer som har en ID.(25 svar)

C) Informasjon om alle land som har en hovedstad.(39 svar)

D) Informasjon om alle kapitalistiske stater.(5 svar)

Her ga jeg nok litt for knapp tid? Ken tenke meg at 20 sek. ble litt lite for å tenke godt over spørringen. Sorry. :-\

Relasjonsalgebra

Relasjonsalgebra

- Relasjonsalgebra er et teoretisk språk.
 - Definert av Codd.
 - SQL er basert på dette.
- Fra emnebeskrivelsen, læringsutbytte:
 - Etter å ha fullført emnet skal studenten kunne:
 - ... beskrive hva relasjonsalgebra er, og forklare mulighetene det gir ved å benytte begrepene **kartesisk produkt, union, snitt og mengde**.

Relasjonsalgebra – forts.

- Relasjonsalgebra (og dermed også SQL) har noen viktige prinsipper:
- Resultatet skal dannes uten å endre kildene (SELECT).
- Resultatet skal følge samme format som kildene.
 - Resultatet fra en operasjon kan dermed være kilden til en annen! (Ref. Subqueries senere.)
- Resultatet av en operasjon omtales gjerne som en mengde.

OBS

- Trykkleif, s 143: «(...) **ariteten** tilsvareer antall tupler.»
- Hva skulle det egentlig ha stått?

Terminologi (repetisjon)

Kolonner (attributter)

<u>FILIAL</u>				
FNR	ADRESSE	STED	LEDERNR	STARTDATO
---	-----	-----	-----	-----
1	UTLEIEVEIEN 15	STAVANGER	3	08-SEP-10
2	SMUGLERSTIEN 67	BERGEN	2	01-JAN-99
3	BEDRAGERIGATEN 66	OSLO	1	01-JUL-94

Rader (tupler)
 Kardinaliteten = 3

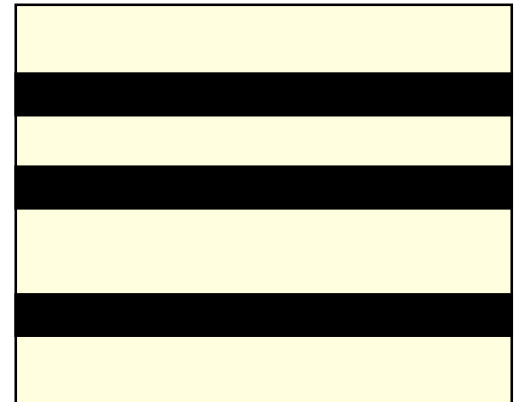
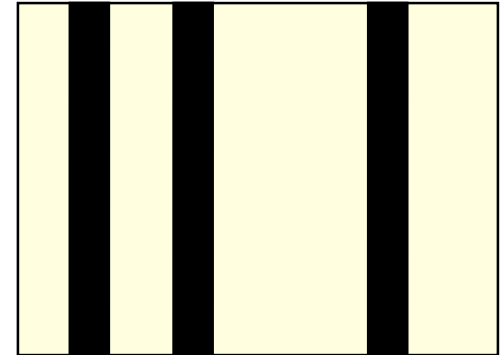
Graden = 5

Domene

Attributt	Domene navn	Betydning	Domene definisjon
Fnr	Filialnummer	Mengden av alle mulige filialnummer	Number (2)
Adresse	Gatenavn	Mengden av alle gater i Norge	Streng (25)
Startdato	Startdato	Mengden av alle datoer	Date, fra 1-JAN-1985 format dd-mon-yy

Seleksjon og projeksjon

- En projeksjon av tabellen (relasjonen) R er en mengde bestående av kolonner fra R .
- En seleksjon av tabellen R er en mengde bestående av rader fra R .



Kartesisk produkt

- Kartesisk produkt operasjonen gir som output mengden som kombinerer hver eneste rad ("tuple") i tabell R med hver eneste rad i tabell S.
 - Formel: $R \times S$

R

a
b

S

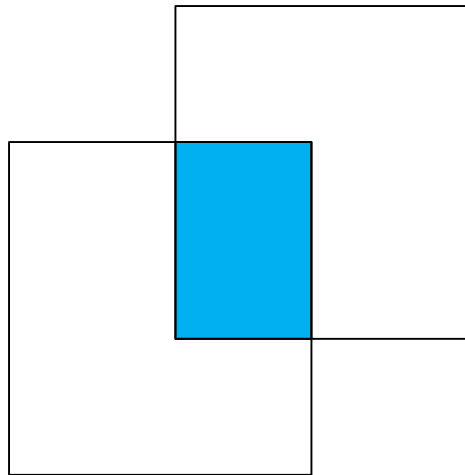
1
2

$R \times S$

a	1
a	2
b	1
b	2

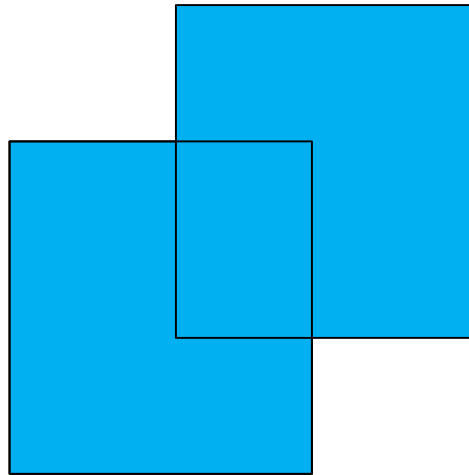
Snitt

- Snittet av R og S er definert som mengden som inneholder radene som finnes i både R og S.
 - Formel: $R \cap S$



Union

- Unionen av R og S er definert som mengden som inneholder radene som finnes i R eller S. (Duplikater tas ikke med.)
 - Formel: $R \cup S$



Union i SQL

Reserverte ord i [MySQL](#):

	UNDEFINED
	UNDO_BUFFER_SIZE
	UNION (R)
	UNLOCK (R)
	UPDATE (R)

	ACTION
	AGAINST
	ALL (R)
	ANALYSE
	ANY

Union i SQL

[w3schools](https://www.w3schools.com/sql/union.asp):

The SQL UNION Operator

The UNION operator is used to combine the result-set of two or more SELECT statements.

Notice that each SELECT statement within the UNION must have the same number of columns. The columns must also have similar data types. Also, the columns in each SELECT statement must be in the same order.

SQL UNION Syntax

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

Note: The UNION operator selects only distinct values by default. To allow duplicate values, use the ALL keyword with UNION.

Union eksempel

- Navn på land og byer som begynner på bokstavene Ar:

```
SELECT Name
FROM country
WHERE Name LIKE 'Ar%';
```

name
Aruba
Argentina
Armenia

```
SELECT Name
FROM city
WHERE Name LIKE 'Ar%';
```

name
Arnhem
Aracaju
Arapiraca
Araraquara
Araçatuba
Araguaína
Araras
Araguari
Arica
Arayat
Arrah (Ara)
Arak
Ardebil
Arezzo
Armenia
Arequipa
Arecibo
Argenteuil
Arad
Araçatuba
Aricha

30 rader, deriblant:

Union eksempel

- Navn på land og byer som begynner på bokstavene Ar:

```
SELECT Name
FROM country
WHERE Name LIKE 'Ar%'
UNION
SELECT Name
FROM city
WHERE Name LIKE 'Ar%';
```

Name
Aruba
Argentina
Armenia
Arnhem
Aracaju
Arapiraca
Araraqu...
Araçá...
Aragua...
Araras
Araguari
Arica
Arayat
Arrah (...)
Arak

31 rader, deriblant:

Union ALL eksempel

- Navn på land og byer som begynner på bokstavene Ar:

```
SELECT Name
FROM country
WHERE Name LIKE 'Ar%'
UNION ALL
SELECT Name
FROM city
WHERE Name LIKE 'Ar%';
```

Name
Aruba
Argentina
Armenia
Arnhem
Aracaju
Arapiraca
Araraqu...
AraÅŕsa...
Aragua...
Araras
Araguari
Arica
Arayat
Arrah (...)
Arak

33 rader, deriblant:

Hvilke verdier er duplikater?

- Kanskje vi kan finne dem ved å ta snittet? (Snitt = INTERSECT)

```
SELECT Name
FROM country
WHERE Name LIKE 'Ar%'
INTERSECT
SELECT Name
FROM city
WHERE Name LIKE 'Ar%';
```

Ser ikke lovende ut...

```
SELECT Name
FROM country
WHERE Name LIKE 'Ar%'
INTERSECT
SELECT Name
FROM city
WHERE Name LIKE 'Ar%';
```

INSERT_METHOD	INSTANCE	INSTANCE
INT (R)	INT1 (R)	INT2 (R)
INT3 (R)	INT4 (R)	INT8 (R)
INTEGER (R)	INTERVAL (R)	INTO (R)
INVOKER	IO	IO_AFTER_GTIDS (R)
IO BEFORE GTIDS (R)	IO THREAD	IPC

Eksempel på variasjon i DBMS'er

- Oracle, [reserveerte ord](#) (utdrag):

```
INDEX  
INITIAL  
INSERT *  
INTEGER *  
INTERSECT *  
INTO *  
IS *  
LEVEL  
LIKE *  
LOCK  
LONG  
MAXEXTENTS  
MINUS  
MLSLABEL  
MODE
```

Men vi får det til...

- Vi ønsker altså å finne navnene som forekommer i både land og by.

```
SELECT Name
FROM country
WHERE Name IN (
  SELECT Name
  FROM city
  WHERE Name LIKE 'Ar%');
```

Name
Armenia

31+1!=33

Kan noen forklare?
Land på Ar (3):

Aruba
Argentina
Armenia

Byer på Ar (30):

Snitt(1):

Name
Armenia

Arnhem
Aracaju
Arapiraca
Araraquara
AraÃ§atuba
AraguaÃna
Araras
Araguari
Arica
Arayat
Arrah (Ara)
Arak
Ardebil
Arezzo
Armenia
Arequipa
Arecibo
Argenteuil
Arad
AraÃ´ar
Arusha
Ariana
Araure
Arkangeli
Armavir
Arzamas
Arlington
Arlington
Arvada
Arden-Arcade

UNION(31):

Aruba
Argentina
Armenia
Arnhem
Aracaju
Arapiraca
Araraquara
AraÃ§atuba
AraguaÃna
Araras
Araguari
Arica
Arayat
Arrah (Ara)
Arak
Ardebil
Arezzo
Arequipa
Arecibo
Argenteuil
Arad
AraÃ´ar
Arusha
Ariana
Araure
Arkangeli
Armavir
Arzamas
Arlington
Arvada
Arden-Arcade

UNION ALL(33):

Aruba
Argentina
Armenia
Arnhem
Aracaju
Arapiraca
Araraquara
AraÃ§atuba
AraguaÃna
Araras
Araguari
Arica
Arayat
Arrah (Ara)
Arak
Ardebil
Arezzo
Armenia
Arequipa
Arecibo
Argenteuil
Arad
AraÃ´ar
Arusha
Ariana
Araure
Arkangeli
Armavir
Arzamas
Arlington
Arlington
Arvada
Arden-Arcade

Altså...

- Vi har direkte støtte for UNION i (My)SQL. Vi kan benytte UNION ALL hvis vi vil beholde duplikater.
- Vi har ikke noe reservert ord for snitt (INTERSECT) i MySQL, men vi kan finne det i andre DBMSer som f.eks Oracle.
- Men vi kan utføre snitt vha blant annet JOIN og IN. Ta gjerne en titt på EXISTS også...

SQL: Søk over flere tabeller

- Hva om vi vil hente ut data fra flere tabeller?

Eier_id	Navn
1	Per Persen
2	Ola Olsen
3	Kari Normann

Bileier

Regnr	Modell	Eier_id
KH22222	Skoda	1
KH33333	Ferrari	NULL
DE22222	Volvo	2

Bil

Jeg vil hente ut bileiers navn + bileiers registrerte biler (registreringsnummer og modell).

Kartesisk produkt

```
SELECT * FROM bileier, bil;
```

Result Grid   Filter Rows: <input type="text"/> Export: 					
	eier_id	navn	regnr	modell	eier_id
▶	1	Per Persen	DE22222	Volvo	2
	2	Ola Olsen	DE22222	Volvo	2
	3	Kari Normann	DE22222	Volvo	2
	1	Per Persen	KH22222	Skoda	1
	2	Ola Olsen	KH22222	Skoda	1
	3	Kari Normann	KH22222	Skoda	1
	1	Per Persen	KH33333	Ferrari	NULL
	2	Ola Olsen	KH33333	Ferrari	NULL
	3	Kari Normann	KH33333	Ferrari	NULL

Sniktitt: Modelling

Sniktitt på modellering

- Vi har alt jobbet mot world databasen og dens 3 tabeller (entiteter).
 - Innholdet (attributtene) per tabell er greie: de ser vi i venstrekolonnen i MySQL Workbench.
- Men hvordan ser egentlig koplingene (relasjonene) mellom tabellene ut?
 - Hvilke relasjoner har vi mellom tabellene country, city og countryLanguage?

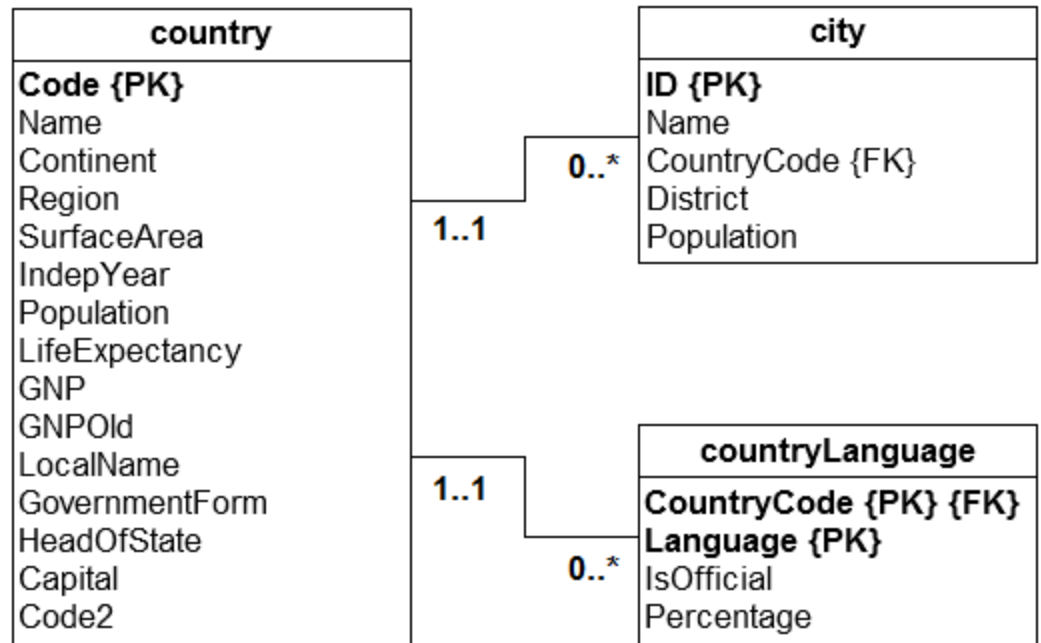
ER modell

- Vi kan vise modellen som et ER diagram:

- Entity

- Relationship

- Modell laget i Gliffy:
<https://www.gliffy.com/>



Neste gang

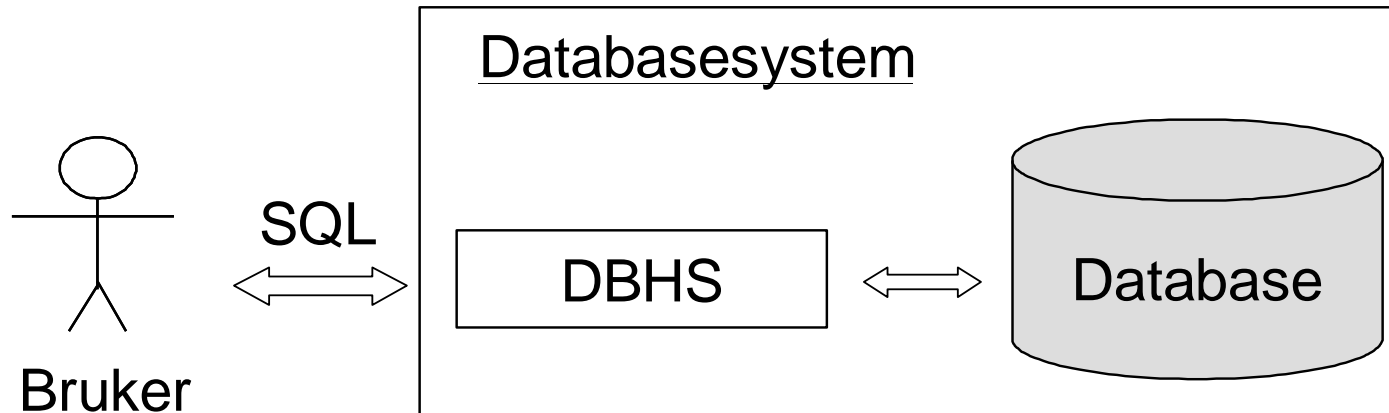
- Neste gang:
 - Kapittel 7: Datamodellering med ER
 - Vi skal benytte [LearnER](#)

Repetisjon

Spm. eller ting dere lurer på?

- Kom gjerne med det nå!
- Evt. kjører jeg det jeg rekker av noen Isides jeg har funnet frem selv.

Database begreper



- SQL (Structured Query Language) er et språk for å kommunisere med relasjons-databaser.

Terminologi

Kolonner (attributter)

<u>FILIAL</u>				
FNR	ADRESSE	STED	LEDERNR	STARTDATO
---	-----	-----	-----	-----
1	UTLEIEVEIEN 15	STAVANGER	3	08-SEP-10
2	SMUGLERSTIEN 67	BERGEN	2	01-JAN-99
3	BEDRAGERIGATEN 66	OSLO	1	01-JUL-94

Rader (tupler)
 Kardinaliteten = 3

Graden = 5

Domene

Attributt	Domene navn	Betydning	Domene definisjon
Fnr	Filialnummer	Mengden av alle mulige filialnummer	Number (2)
Adresse	Gatenavn	Mengden av alle gater i Norge	Streng (25)
Startdato	Startdato	Mengden av alle datoer	Date, fra 1-JAN-1985 format dd-mon-yy

Relasjonsdatabaser – forts.

Primærnøkkel i
Person tabellen



Person:

Pers_ID	Surname	First_Name	City
0	Miller	Paul	London
1	Ortega	Alvaro	Valencia
2	Huber	Urs	Zurich
3	Blanc	Gaston	Paris
4	Bertolini	Fabrizio	Rom

Merk: Alle verdier i en kolonne må være av samme datatype. (Tekst, heltall, desimaltall, ...)

Car:

Car_ID	Model	Year	Value	Pers_ID
101	Bentley	1973	100000	0
102	Rolls Royce	1965	330000	0
103	Peugeot	1993	500	3
104	Ferrari	2005	150000	4
105	Renault	1998	2000	3
106	Renault	2001	7000	3
107	Smart	1999	2000	2

Rad →

Primærnøkkel
i Car tabellen

Kolonne

Fremmednøkkel
i Car tabellen

Relasjonsmodellen: Terminologi

- En relasjon er en tabell med kolonner og rader.
 - En tuppel er et annet navn for en rad i tabellen.
 - En attributt er en navngitt kolonne i tabellen.
 - Et domene er mengden tillatte verdier for et eller flere attributter.
- Vi sier noe om størrelsen til en tabell (relasjon) ut i fra:
 - Graden til en tabell: antall kolonner den inneholder.
 - Kardinaliteten til en tabell: antall rader den inneholder.
- En relasjonsdatabase er en samling relasjoner.
 - Nivået av strukturering angir normaliseringen (kommer tilbake til normalisering på en senere forelesning).

Egenskaper for å være en tabell

- Hver tabell må ha et unikt navn (innenfor denne database / dette schema).
- Hver kolonne må ha et unikt navn (innenfor denne tabell).
 - Verdien til en kolonne må være fra samme domene.
 - Rekkefølgen på kolonnene kan ikke ha betydning.
- Det skal ikke finnes like rader (alle er unike).
 - Rekkefølgen på radene kan ikke ha betydning.
- Hver celle skal inneholde én gyldig verdi fra kolonnens domene (eller NULL – hvis tillatt).

SQL

- select, insert, update, delete, mm
- DDL
- join
- subquery
- datatyper
- funksjoner

Noen SQL funksjoner

- SQL har noen innebygde funksjoner, bl.a.:
 - COUNT(*) → antall
 - AVG(kolonne_navn) → gjennomsnitt
 - SUM(kolonne_navn) → sum
 - MIN(kolonne_navn) → minimum
 - MAX(kolonne_navn) → maksimum
- Det var disse vi var innom først. Men vi har også benyttet:
 - STR_TO_DATE(str, format)
 - DATE_FORMAT(date, format)

Verdien NULL

- NULL representerer en kolonneverdi som ikke er satt for denne raden i tabellen.
- MERK:
 - NULL er ikke det samme som tallet 0.
 - NULL er ikke det samme som en blank/space.

	Code	Name	IndepYear
▶	ABW	Aruba	NULL
	AFG	Afghanistan	1919
	AGO	Angola	1975
	AIA	Anguilla	NULL
	ALB	Albania	1912
	AND	Andorra	1278
	ANT	Netherlands Antilles	NULL
	ARE	United Arab Emirates	1971
	ARG	Argentina	1816
	ARM	Ameria	1991
	ASM	American Samoa	NULL
	ATA	Antarctica	NULL
	ATF	French Southern territories	NULL

Dat typer

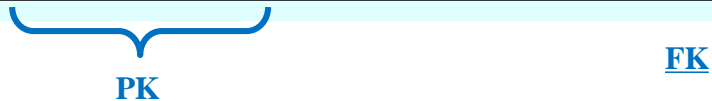
- (Navn og syntaks for datatyper varierer litt fra database til database.)
- MySQL inneholder en rekke datatyper. Blant de vanligere er: char, varchar, int, float, date og enum.
- Fullstendig oversikt (for både MySQL og andre, kjente relasjonsdatabaser) finner dere bl.a. på: [SQL datatypes @ w3schools.com](http://www.w3schools.com/sql/datatypes.asp).

Nøkler

- Primærnøkkel (PK – primary key) er den unike identifikatoren for radene i en tabell. PK kan være sammensatt av flere kolonner.
- Fremmednøkkel (FK – foreign key) er en (eller flere) kolonne(r) i en tabell som viser til (har samme verdi som) en primærnøkkel i en annen (evt. samme) tabell.
- (Supernøkler og kandidatnøkler var lesestoff til i dag...)

<u>LEIEKONTRAKT</u>				
<u>EIENDOMNR</u>	<u>FRADATO</u>	<u>TILDATO</u>	<u>LNR</u>	<u>LEIE</u>

1	01-JUN-05	31-MAY-10	3	6500
1	01-JUN-08		3	7500
2	01-AUG-98	31-JAN-06	4	11000



PK FK

Oppgaver (rep)

Eier:

Id	Navn	Adresse	Tlf	Epost
1	Ola Olsen	Liksomveien 2	22222222	ola@online.no
2	Ola Olsen	Januarveien 2	33333333	ola@is.com
3	Ina Jensen	Juliveien 3	44444444	ina@is.com

Husdyr:

Id	Dyr	Navn	Eier
1	Katt	Mia	2
2	Hund	Passop	2
3	Papegøye	Polly	1
4	Katt	Kitty	3

1. Hvilke tabeller har vi?
2. Hvilken kardinalitet og grad har tabellene?
3. Hvilke kolonner kan være primærnøkler?
4. Hvilke kolonner kan være fremmednøkler?

Oppgaver (rep)

City

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e...	AFG	Balkh	127800
5	Amster...	NLD	Noord-H...	731200
6	Rotterd...	NLD	Zuid-Holl...	593321
7	Haag	NLD	Zuid-Holl...	440900
8	Utrecht	NLD	Utrecht	234323

Countrylanguage

CountryCode	Language	IsOfficial	Percentage
ABW	Dutch	T	5.3
ABW	English	F	9.5
ABW	Papiamentu	F	76.7
ABW	Spanish	F	7.4
AFG	Balochi	F	0.9
AFG	Dari	T	32.1
AFG	Pashto	T	52.4
AFG	Turkmenian	F	1.9

Country

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOld	LocalName	GovernmentForm	HeadOfState	Capital	Code2
ABW	Aruba	North America	Caribbean	193.00	NULL	103000	78.4	828.00	793.00	Aruba	Nonmetropolitan T...	Beatrix	129	AW
AFG	Afghani...	Asia	Souther...	652090.00	1919	22720000	45.9	5976.00	NULL	Afganistan/A...	Islamic Emirate	Mohammad Omar	1	AF
AGO	Angola	Africa	Central ...	1246700.00	1975	12878000	38.3	6648.00	7984.00	Angola	Republic	JosÃ© Eduard...	56	AO
AIA	Anguilla	North America	Caribbean	96.00	NULL	8000	76.1	63.20	NULL	Anguilla	Dependent Territor...	Elisabeth II	62	AI
ALB	Albania	Europe	Souther...	28748.00	1912	3401200	71.6	3205.00	2500.00	ShqipÃ«ria	Republic	Rexhep Mejdani	34	AL
AND	Andorra	Europe	Souther...	468.00	1278	78000	83.5	1630.00	NULL	Andorra	Parliamentary Copri...		55	AD
ANT	Netherl...	North America	Caribbean	800.00	NULL	217000	74.7	1941.00	NULL	Nederlandse...	Nonmetropolitan T...	Beatrix	33	AN

1. Hvilke kolonner kan være primærnøkler?
2. Hvilke kolonner kan være fremmednøkler?

Select - distinct

- Et select-utvalg for et begrenset antall kolonner kan gi like rader i svaret (fordi unike kolonner for disse radene er fjernet).
- For å fjerne evt. duplikater ved select:

```
SELECT DISTINCT CountryCode  
FROM city  
ORDER BY CountryCode ASC
```

where

- Operatorer:

<code>=</code>	lik (<i>ikke</i> v/wildcards!)
<code><></code> eller <code>!=</code>	forskjellig fra
<code><</code>	mindre enn
<code>></code>	større enn
<code><=</code>	mindre eller lik
<code>>=</code>	større eller lik

<code>like</code>	lik, godtar wildcards
<code>in</code>	i gitt utvalg
<code>between</code>	utvalg, følges av <code>and</code>
<code>_</code>	wildcard, enkelt tegn
<code>%</code>	wildcard, flere tegn
<code>is null</code>	evt. <code>is not null</code>

- Logiske operatorer – setter sammen kriterier:

<code>and</code>	og
<code>or</code>	eller
<code>not</code>	ikke

limit

- Begrenser antall rader i resultatet ditt.
- Eksempel:
Jeg vil finne de tre største landene i verden.

```
SELECT Name, SurfaceArea  
FROM Country  
ORDER BY SurfaceArea DESC  
LIMIT 3;
```

Result Grid			Filter Rows:
	Name	SurfaceArea	
▶	Russian Federation	17075400.00	
	Antarctica	13120000.00	
	Canada	9970610.00	
*	NULL	NULL	

order by

- Sorterer utvalget, basert på kolonnenavn.
 - Kan også benytte kolonnennummer, men dette er "deprecated". (← Utgått standard, men fortsatt støttet. ->Vi bruker det altså IKKE...)
- Kan sortere flere kolonner (kommaseparert).
- Rekkefølge styres med asc (ascending - stigende) og desc (descending - synkende).
 - Uten asc eller desc blir rekkefølgen stigende.

group by og having

- group by lar oss gruppere summerings-resultater til mer enn én rad.
- Summeringsresultater får vi når vi bruker funksjoner som count, sum, avg, ...
- Ønsker vi i tillegg å fjerne rader, bruker vi ikke where, men having.

group by og having – Oppgave!

- Hent ut informasjon om de distriktene som har minst 50 byer. Du kan forutsette at navn på distrikter er unike innenfor et land, men ikke på tvers av land...

Hent ut: Landskode, distriktets navn, antall byer og samlet innbyggertall i byene. Hint: Jeg får 3 rader...

Subqueries

- Som nevnt, er resultatet av en SELECT formatert som en ny tabell:
 - Det danner kolonner og rader på samme måte som databasens eksisterende tabeller.
- Derfor er det ikke noe problem å bruke resultatet av en SELECT som et element i en annen!
- Å putte en SELECT inne i en annen på denne måten kalles en subquery.

Oppgave

- Hent ut navn på alle byer som er hovedsteder (vha subquery). Hint: kolonnen «Capital» i Country-tabellen angir hovedstad...

```
SELECT name
FROM City
WHERE id IN(?);
```

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabol	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e...	AFG	Balkh	127800
5	Amster...	NLD	Noord-H...	731200
6	Rotterd...	NLD	Zuid-Holl...	593321
7	Haag	NLD	Zuid-Holl...	440900
8	Utrecht	NLD	Utrecht	234323

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOLD	LocalName	GovernmentForm	HeadOfState	Capital	Code2
ABW	Aruba	North America	Caribbean	193.00	NULL	103000	78.4	828.00	793.00	Aruba	Nonmetropolitan T...	Beatrix	129	AW
AFG	Afghani...	Asia	Souther...	652090.00	1919	22720000	45.9	5976.00	NULL	Afganistan/A...	Islamic Emirate	Mohammad Omar	1	AF
AGO	Angola	Africa	Central ...	1246700.00	1975	12878000	38.3	6648.00	7984.00	Angola	Republic	JosÃ© Eduard...	56	AO
AIA	Anguilla	North America	Caribbean	96.00	NULL	8000	76.1	63.20	NULL	Anguilla	Dependent Territor...	Elisabeth II	62	AI
ALB	Albania	Europe	Souther...	28748.00	1912	3401200	71.6	3205.00	2500.00	ShqipÃ«ria	Republic	Rexhep Mejdani	34	AL
AND	Andorra	Europe	Souther...	468.00	1278	78000	83.5	1630.00	NULL	Andorra	Parliamentary Copri...		55	AD
ANT	Netherl...	North America	Caribbean	800.00	NULL	217000	74.7	1941.00	NULL	Nederlandse...	Nonmetropolitan T...	Beatrix	33	AN

Løsning

```
SELECT name  
FROM City  
WHERE id IN(SELECT capital FROM Country);
```

- Du finner videoer med oppgaver og løsning med subqueries i [spillelisten](#) til emnet..

Join

- Hva om vi ønsker å vise både navnet på landet (Country tabellen) og prosent-andelen i landet som snakker spansk (CountryLanguage tabellen) samtidig?
- Dette kan vi få til ved å join'e flere tabeller sammen under spørringen!

Join – forts.

- Eksempel:

```
SELECT Name, Percentage  
FROM country  
→ JOIN countrylanguage ON Code = CountryCode  
WHERE Language = 'spanish' AND IsOfficial = 'T'  
ORDER BY Name;
```

- Denne type join kalles også inner join.

Left Join og Right Join

- (Inner) join tar bare med resultater som matches i alle involverte tabeller.
- Det finnes to andre varianter av join også:
 - Left join og right join (felles kalt outer join).
- Dette er varianter som tar med rader som ikke matches i alle involverte tabeller.

Left Join og Right Join – forts.


- Eks.: Vi ønsker å hente ut landene som begynner på 'An', og byene i dem.

```
select country.Name, city.Name  
from Country  
join City on Code = CountryCode  
where country.Name like 'An%'  
order by country.Name;
```

- Men henter denne alle land på 'An%'?
 - Tja, time will show! Eller neste slide da ... :-)

Left Join og Right Join – forts.

- Antarctica mangler, fordi det ikke inneholder noen byer!
 - For å få med Antarctica må vi gjøre en LEFT JOIN: (eller "LEFT OUTER JOIN" om du vil)



```
SELECT country.Name, city.Name  
FROM Country  
LEFT JOIN City ON Code = CountryCode  
WHERE country.Name like 'An%'  
ORDER BY country.Name;
```

Left Join og Right Join – forts.

- En LEFT JOIN tar med rader fra den første/venstre tabellen (Country) som ikke finnes i den siste/høyre tabellen (City).
- Tilsvarende har vi RIGHT JOIN (eller "RIGHT OUTER JOIN") som tar med rader fra den siste/høyre tabellen som ikke finnes i den første/venstre.
- (Merk: Det finnes andre typer outer join og, men left/right er de vanlige, og de som er viktige (pensum) ifbm. DB1100.)

Join – oversikt

- Disse skal dere kunne:

Vi skriver	Kan også skrives
JOIN	INNER JOIN
LEFT JOIN	LEFT OUTER JOIN
RIGHT JOIN	RIGHT OUTER JOIN

Oppgave (rep)

id	name	email
123	Bart	bart@fox.com
456	Milhouse	milhouse@fox.com
888	Lisa	lisa@fox.com
404	Ralph	ralph@fox.com

students

id	name	teacher_id
10001	Computer Science 142	1234
10002	Computer Science 143	5678
10003	Computer Science 190M	9012
10004	Informatics 100	1234

courses

student_id	course_id	grade
123	10001	B-
123	10002	C
456	10001	B+
888	10002	A+
888	10003	A+
404	10004	D+

grades

id	name
1234	Krabappel
5678	Hoover
9012	Stepp

teachers

[Hentet fra University of Washington](#)

- Lag en spørring (med join) som henter ut navn på alle lærere og navn på alle kurs de underviser i.

```
SELECT teachers.name, courses.name
FROM teachers LEFT JOIN courses
ON teachers.id = courses.teacher_id;
```

SQL DDL m.m: endring tabeller

- DDL = Data Definition Language. Del av SQL. Vi har følgende SQL kommandoer for å modifisere tabeller:
 - CREATE TABLE
 - ALTER TABLE
 - DROP TABLE
- Og følgende SQL kommandoer for å modifisere innhold: (NB – ”vanlig” SQL, ikke del av DDL)
 - INSERT INTO
 - UPDATE
 - DELETE FROM

Videre arbeid i dag, og kort om neste gang

- Neste gang, nytt tema: **Modellering av databaser** (torsdag om 1 uke)
 - "Flipped" struktur, 2t øving 12:15 – 14:00, 2t teori 14:15 til 16:00.
 - Pensum: **Kapittel 7** – Datamodellering med ER.
- Snart: 2t multiple choice prøve for å teste deg selv.
 - Finnes på Canvas når øving starter, fasit legges der mot slutten av øvingstiden. (Ingen karakter, skal ikke leveres inn – for å teste deg selv.)
 - Evt. prioriter å jobbe med oppgaver fra tidligere økter.
- **NÅ: Kahoot!**

Kahoot - hva husker vi fra dagens innhold?

- <https://kahoot.it/#/>
 - Eller gjerne Kahoot appen for mobil.
- Dagens **12** spm. er hentet fra tidligere eksamener!
- **Fortsetter dere å slå fjorårets studenter? :-)**
- Fjorårets score: 64,17 %

Overall Performance	
Total correct answers (%)	64,17%
Total incorrect answers (%)	35,83%
Average score (points)	7481,00 points