

# SQL: spørringer med JOIN

Modul 9 (uke 7) innhold

# SQL: Søk over flere tabeller

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

Bileier	
Eier_id	Navn
1	Per Persen
2	Ola Olsen
3	Kari Normann

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

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

# Kartesisk produkt og SQL

- *IKKE* ok oppsett:

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

- Kartesisk produkt!
  - For mange rad-kombinasjoner returnert.

Result Grid					
		Filter Rows:		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

# Kartesisk produkt

- Kartesisk produkt operasjonen gir som output mengden som kombinerer hver eneste rad i tabell R med hver eneste rad i tabell S.

- Formel:  $R \times S$

R	S	R x S	
a	1	a	1
b	2	a	2
		b	1
		b	2

- Kartesisk produkt:
  - Vi plusser sammen kolonner, ganger sammen rader.
- Så for World, med: `SELECT * FROM country, city;` får vi da ...?
  - Knappe 1 000 000 rader i svaret! O\_o
  - (Fun fact: Går fort! Å hente ut 974 881 rader tar knappe 2. sek på min maskin.)

# Join

- Riktig SQL oppsett:

```
SELECT *  
FROM bileier  
JOIN bil ON bileier.eier_id = bil.eier_id;
```

Result Grid						Filter Rows:	Export:
	eier_id	navn	regnr	modell	eier_id		
▶	2	Ola Olsen	DE22222	Volvo	2		
	1	Per Persen	KH22222	Skoda	1		

- Slår sammen tabellene basert på kolonnen med PK-FK innhold (eier\_id).
  - Vi henter kun de radene som har felles eier\_id.
  - SQL syntaks for dette: **JOIN**

# Join

- **JOIN** gir oss alle kolonner, men kun for de relevante, **sammenhengende radene fra begge tabeller**.
  - Vi kan f.eks. hente ut "bileiers navn, reg.nummer og modell":  
(dataene er fordelt på bileier og bil tabellene)

```
SELECT navn, regnr, modell  
FROM bileier  
JOIN bil ON bileier.eier_id = bil.eier_id;
```

Result Grid			
Filter Rows:			
	navn	regnr	modell
▶	Ola Olsen	DE22222	Volvo
	Per Persen	KH22222	Skoda

# Left join

- Men hva om vi også vil ha med personer som ikke eier noen bil?
  - Benytter da **LEFT JOIN**: (med bileier som venstre tabell)

```
SELECT bileier.eier_id, navn, regnr, modell
FROM bileier
LEFT JOIN bil ON bileier.eier_id = bil.eier_id;
```

	eier_id	navn	regnr	modell
▶	2	Ola Olsen	DE22222	Volvo
	1	Per Persen	KH22222	Skoda
	3	Kari Normann	NULL	NULL

- Vi tar **utgangspunkt i venstre tabell** (bileier) og tar med relevant data fra **høyre tabell** (bil) **der det finnes en kopling** mellom tabellene.
  - Med LEFT JOIN får vi med Kari Normann, selv om hun ikke eier en bil.

# Right join

- Hva om vi heller vil ha med alle biler, også de uten eier?
  - Da benytter vi **RIGHT JOIN**: (med bileier som venstre tabell)

```
SELECT bileier.eier_id, navn, regnr, modell
FROM bileier
RIGHT JOIN bil ON bileier.eier_id = bil.eier_id;
```

	eier_id	regnr	modell	navn
▶	1	KH22222	Skoda	Per Persen
	2	DE22222	Volvo	Ola Olsen
	NULL	KH33333	Ferrari	NULL

- Vi tar **utgangspunkt i høyre tabell** (bil) og inkluderer data fra venstre tabell (bileier) der det finnes en kopling for disse.
  - Med RIGHT JOIN får vi med KH33333 Ferrari, selv om den ikke har noen eier.



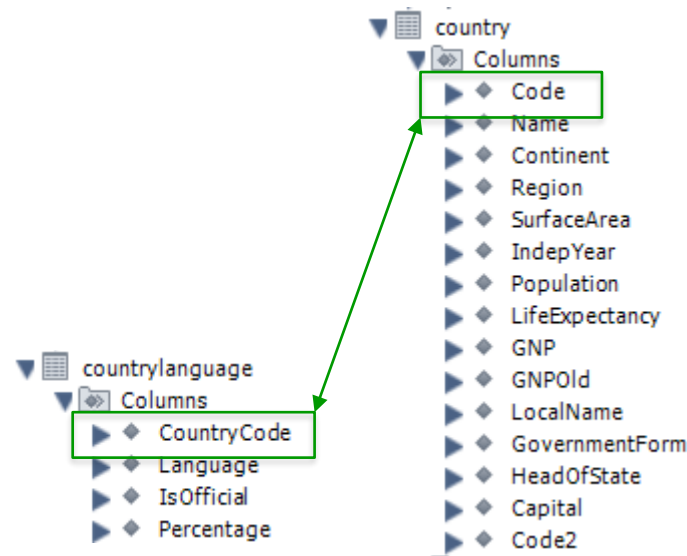
# SQL: Søk over flere tabeller

- World-databasen: Vi ønsker å finne hvilke land som har spansk som offisielt språk.
- Fra før av har vi kunnskap nok til å gjøre dette mot countrylanguage tabellen.
  - Kan da imidlertid ikke få ut navnet på landet, bare landkodene:

```
SELECT CountryCode, Percentage  
FROM countrylanguage  
WHERE Language = 'Spanish' AND IsOfficial = 'T'  
ORDER BY CountryCode;
```

# JOIN

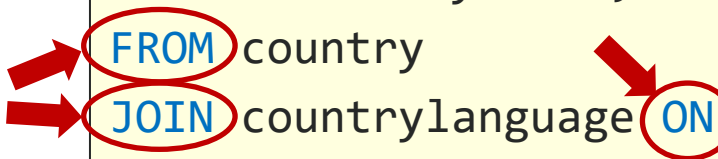
- Hva om vi ønsker å vise både navnet på landet (Country-tabellen) og prosentandelen i landet som snakker spansk (CountryLanguage-tabellen) samtidig?
  - Vi kan bruke join! :-)



# JOIN

- Eksempel:

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



- Denne type **JOIN** kalles også *inner join*.
- *Merk:* Vi trenger ikke spesifisere tabellnavnene i SELECT eller ON delene så lenge kolonnene har unike navn på tvers av tabellene.

# JOIN

- 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 Country.Code = City.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 ... :-P

# 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:

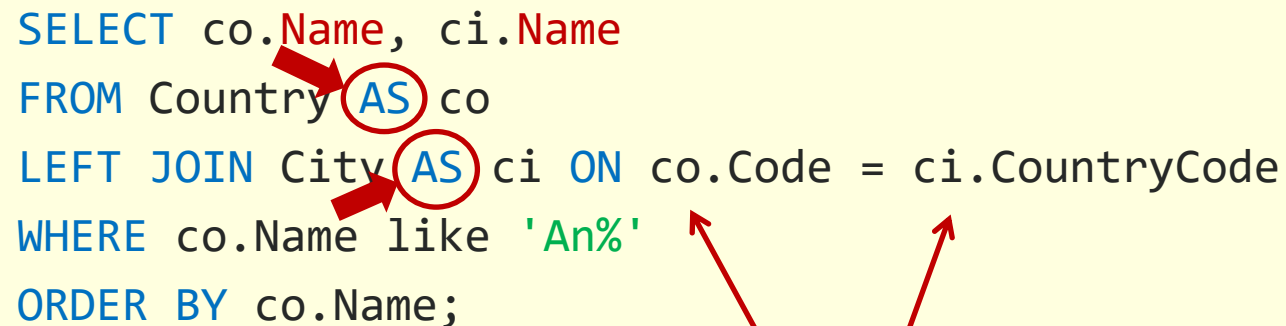
```
SELECT country.Name, city.Name
FROM Country
LEFT JOIN City ON Country.Code = City.CountryCode
WHERE country.Name like 'An%'
ORDER BY country.Name;
```

- *Igjen:* LEFT JOIN tar med rader fra den første/venstre tabellen (Country) som ikke finnes i den siste/høyre tabellen (City).

# Noen tips for Join

- Vi må bruke tabell-prefix for å skille identiske kolonnenavn i flere tabeller fra hverandre.
  - *MERK!* Det er lov å bruke alias også her: (og kan droppe selve ordet **AS**)

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



- Det er ikke nødvendig med tabell-prefix (eks: co.Code) dersom kolonnen kun er definert i én av tabellene. (Men du vil oppnå bedre ytelse med det.)

# Join – oversikt

- Disse skal dere kunne:

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

# Join / kartesisk produkt med betingelse

- Det er veldig vanlig å hente ut data fra flere tabeller, og som vi nå har sett på har vi også en egen syntaks for å gjøre dette: **JOIN**
- Vi kan skrive dette uten å bruke ordet JOIN om vi foretrekker det.
  - Her er to eksempler som gir samme resultat, men kun det første bruker JOIN:

```
SELECT ci.Name, ci.Population, co.name
FROM Country co
JOIN City ci ON co.Code = ci.CountryCode
WHERE co.continent = 'Europe';
```

```
SELECT ci.Name, ci.Population, co.name
FROM Country co, City ci
WHERE co.Code = ci.CountryCode
      AND co.continent = 'Europe';
```



