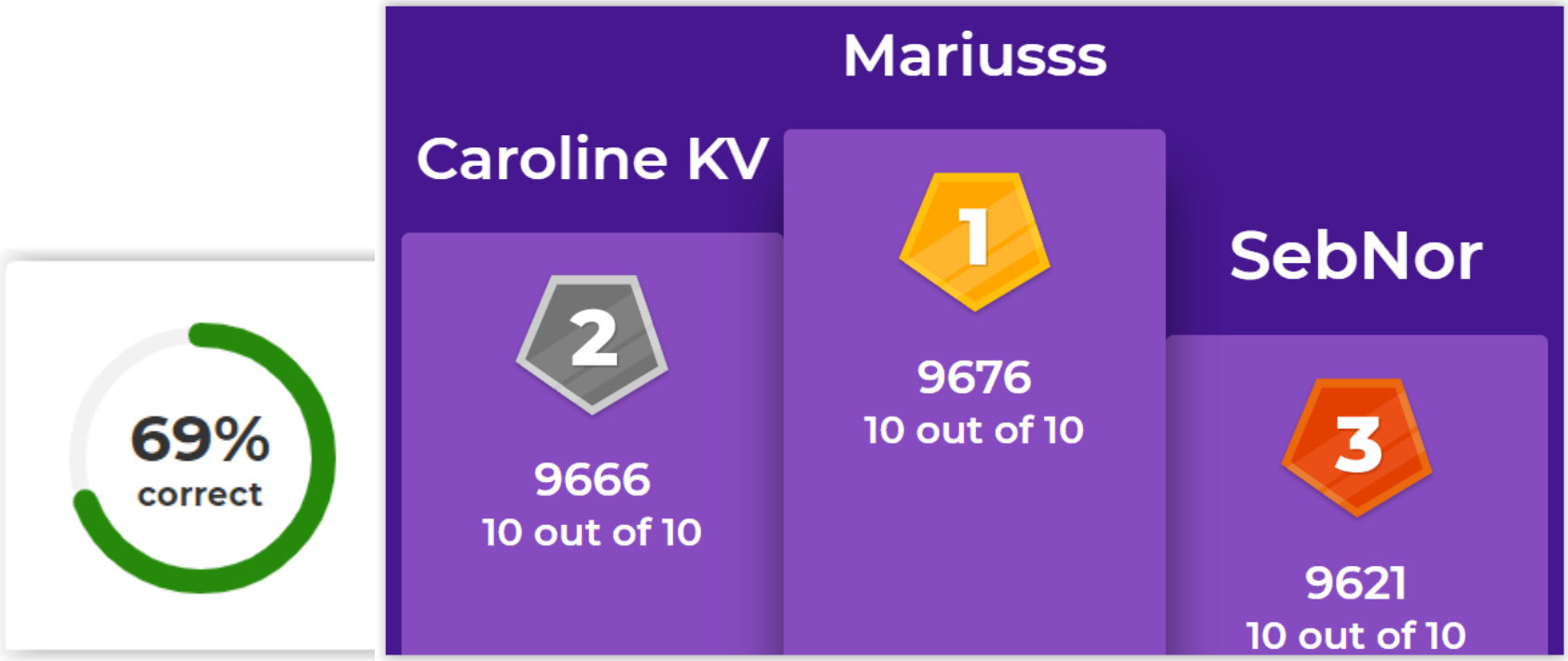
Økt 2 (av 12)

DB1102 Databaser

Per Lauvås / per.lauvas@kristiania.no

Yuan Lin / yuan.lin@kristiania.no

Kahoot


69%
correct
Mariusss
1
9676
10 out of 10
Caroline KV
2
9666
10 out of 10
SebNor
3
9621
10 out of 10

Dagens tema

- Dagens tema: Spørringer mot én tabell.
  - Dagens pensum: Læreboka, kapittel 2.

- Vi kjører samme opplegg som i forrige økt (og hver uke fremover).
  - Forberedelse fram til øving
  - Øving onsdag (sjekk ditt tidspunkt og sted i TimeEdit)
  - Samling over Zoom kl 13.15

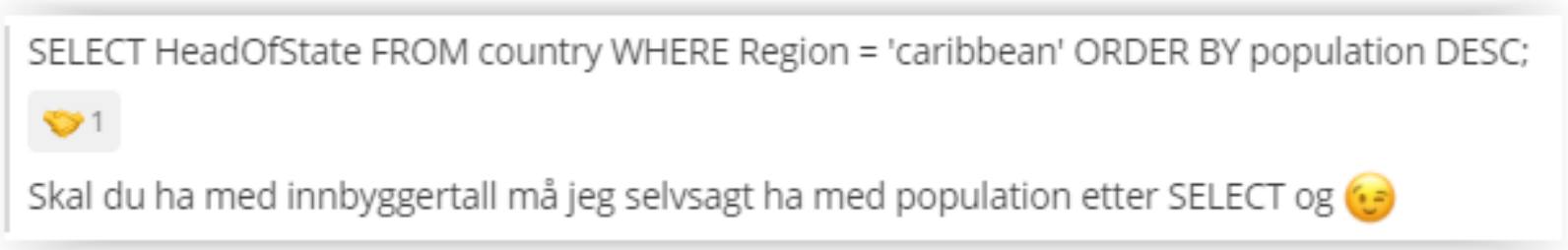
Løsningsforslag øvingsoppgaver forrige økt

- Se løsningsforslag på Canvas.
  - Du finner alltid skriftlige løsningsforslag på Canvas. De publiseres rett etter øvingen.

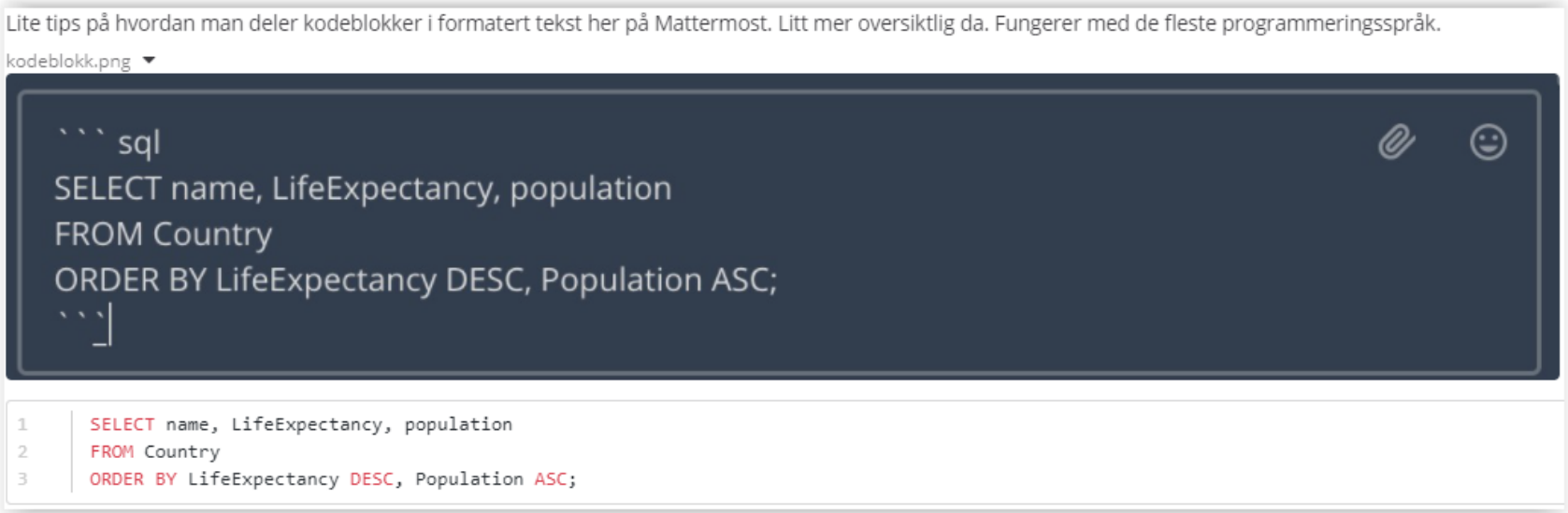
- Til noen av løsningsforslagene finnes også videoer.
  - Du finner dem i emnets spilleliste. (Link også på Canvas.)

Om bruken av Mattermost i forrige økt

- Mange gode oppgaver ble delt på Mattermost!
  - Ikke få panikk hvis du syntes noen var vanskelige:-)



Og vi fikk noen gode tips

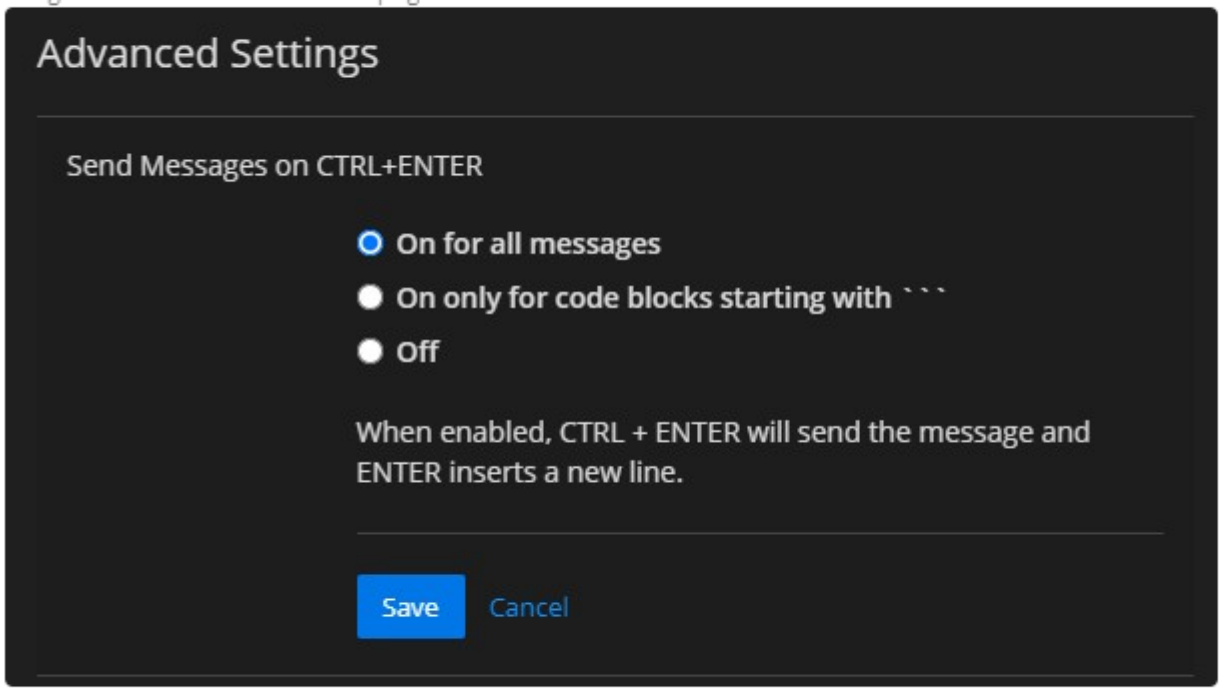


https://docs.mattermost.com/messaging/formatting-text.html

Tips 2

Anbefaler på det sterkeste å slå dette på under advanced settings i account settings, gjør det litt enklere å skrive kode her.
(edited)

Image Pasted at 2021-8-25 18-39.png



3 1

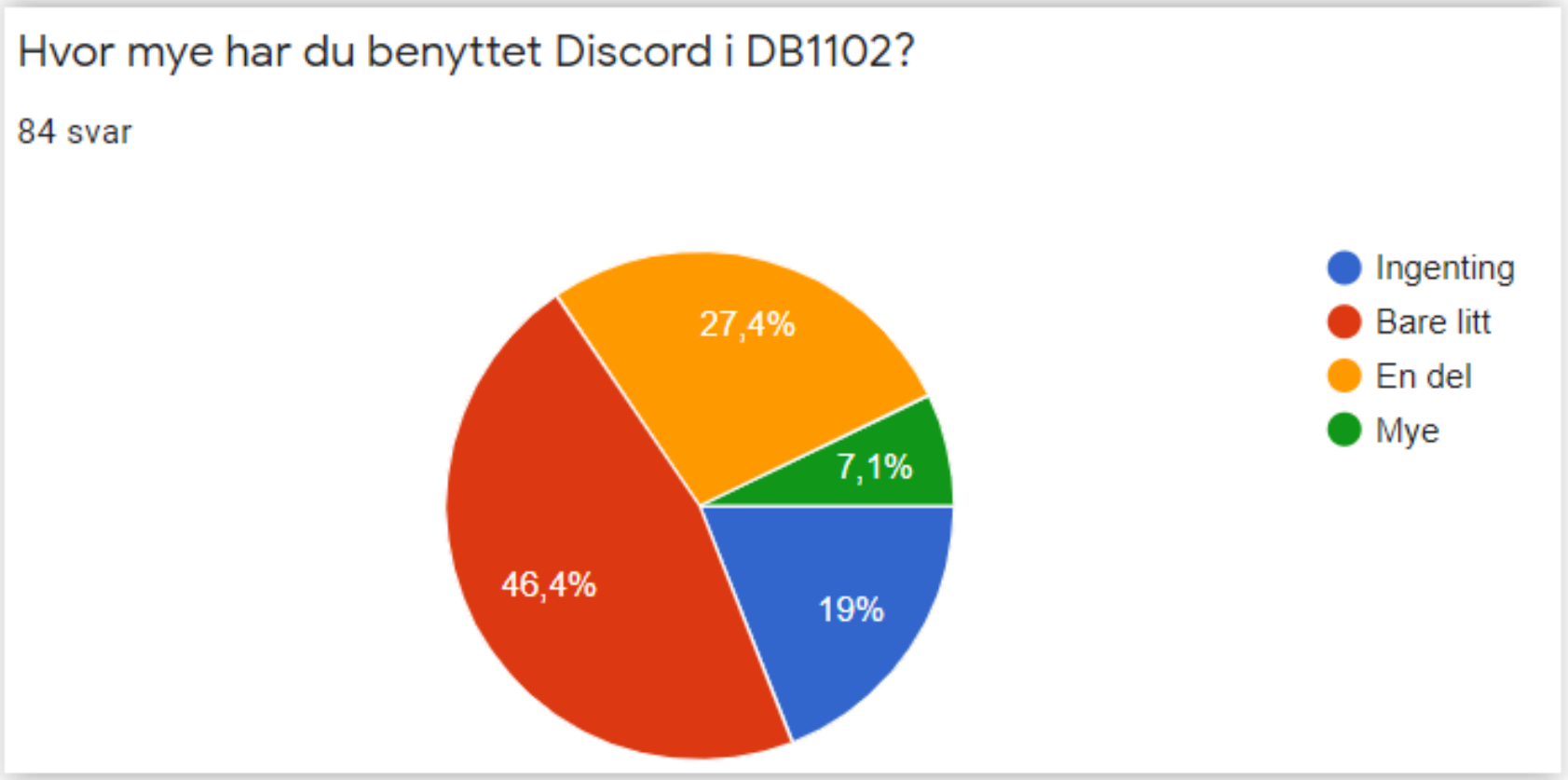
Så slipper man å trykke shift+enter for å lage nye linjer når man skriver i chatten her.

Noen flere eksempler på oppgaver

- Middels: Finn den gjennomsnittlige levealderen i Norden. Forventet resultat: 78.3
- Middels: Ranger land i Europa etter areal økende
- Hent alle byer fra Danmark, Finland, Færøyene, Grønland, Island, Norge, og Sverige. Sorter etter befolkning i en stigende rekkefølge. Kall denne listen for NordiskeByer. Forventet resultat; To kolonner, en som heter Nordiskebyer der Nuuk er øverst og Stockholm nederst og en som heter population.

Takk til alle som bidrar! Fortsett med det☺

Bruk av Discord i fjor



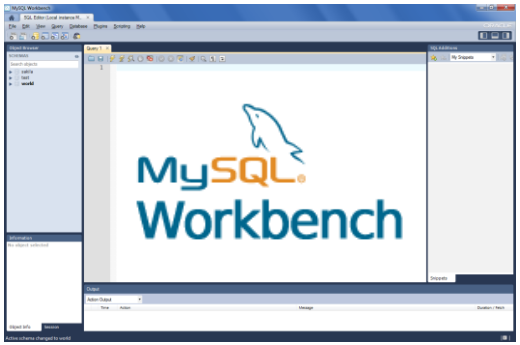
Hvorfor bare litt?

- Følte jeg ikke hadde behov for å benytte det mer enn det jeg gjorde
- Sjenert
- Føler meg ikke helt komfortabel med å bruke discord enda. Det er veldig mye nytt for meg.

Databasebegreper og MySQL (rep. fra forrige økt)

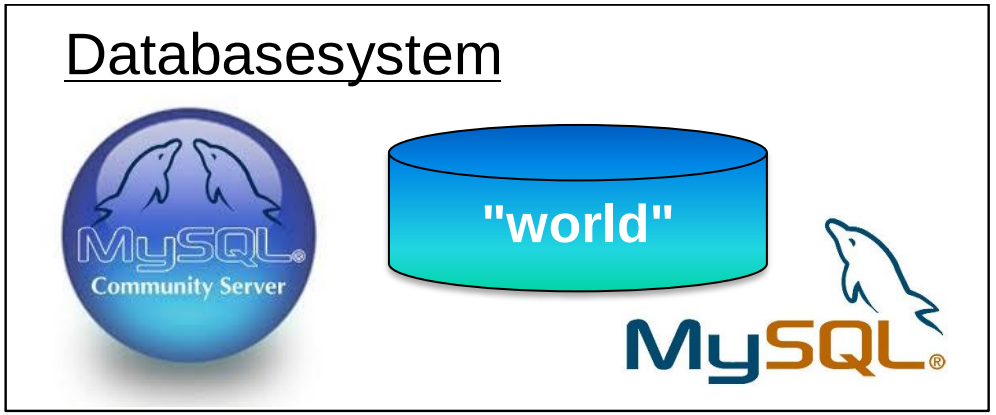
- Databasesystem = DBMS + database.
- DBMS = DataBase Management System:
- Vårt databasesystem: MySQL Community Server + Schema ("world").

- MySQL Workbench blir da ...?
  - Klienten (brukeren) som kommuniserer med DBMS.



Bruker

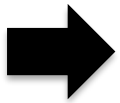
SQL



Spørreresultat – eksempel (rep. fra forrige økt)

- En utvalgsspørring (SELECT query) tar en eller flere tabeller som ”inndata” og gir som ”utdata” et resultat som også er på ”tabellform”.

```
SELECT Name, Population
FROM city
WHERE CountryCode = 'NOR'
ORDER BY Population DESC;
```



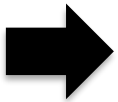
Name	Population
Oslo	508726
Bergen	230948
Trondheim	150166
Stavanger	108848
Bærum	101340

Noen SQL funksjoner (rep. fra forrige økt)

- 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

- For å få en meningsfull overskrift for slike kolonner kan vi gi resultatene egne navn. Dette kan gjøres ved å bruke det reserverte SQL ordet AS.

```
SELECT COUNT(*) AS 'Antall byer'
FROM City;
```



Antall byer
4079

Hva er et db-script?

- Noen SQL-er i en fil som er kjørbar i en database.

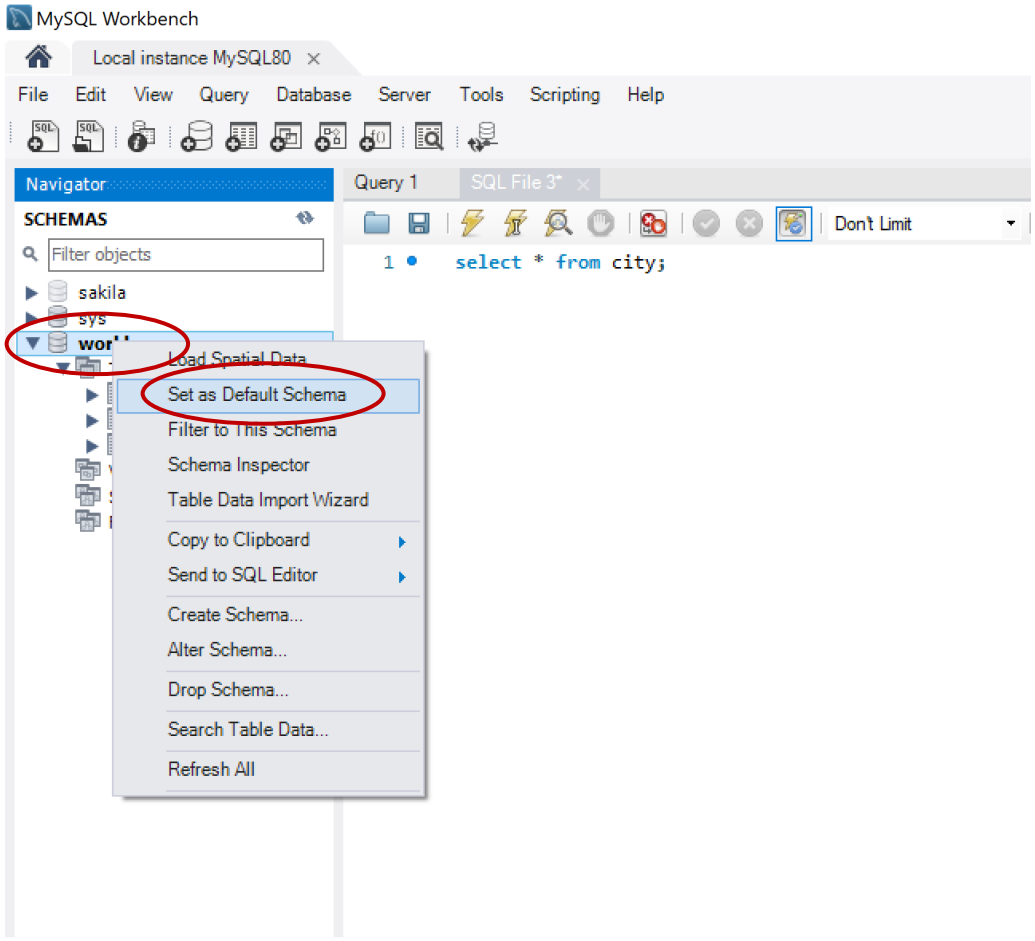
- Jeg legger ut db-script til en del forelesninger, slik at dere enkelt kan prøve ut SQL-ene dere ser på slidene.
  - Alternativt kan dere skrive av SQL fra slides for hånd. (Bedre læring av å skrive selv?)
  - Slike script ligger sammen med den aktuelle forelesningen, i Canvas.

- Kjør den relevante SQL-en for å se at det funker. Endre gjerne litt og se om resultatet blir som antatt.
  - Å kjøre en SQL selv gir bedre læring enn å se en SQL!

- Du finner mange db-script på pensumbokas datasettside på nett!
  - (Link til boka finnes også på: Canvas -> DB1102 -> "Om emnet" modulen.)

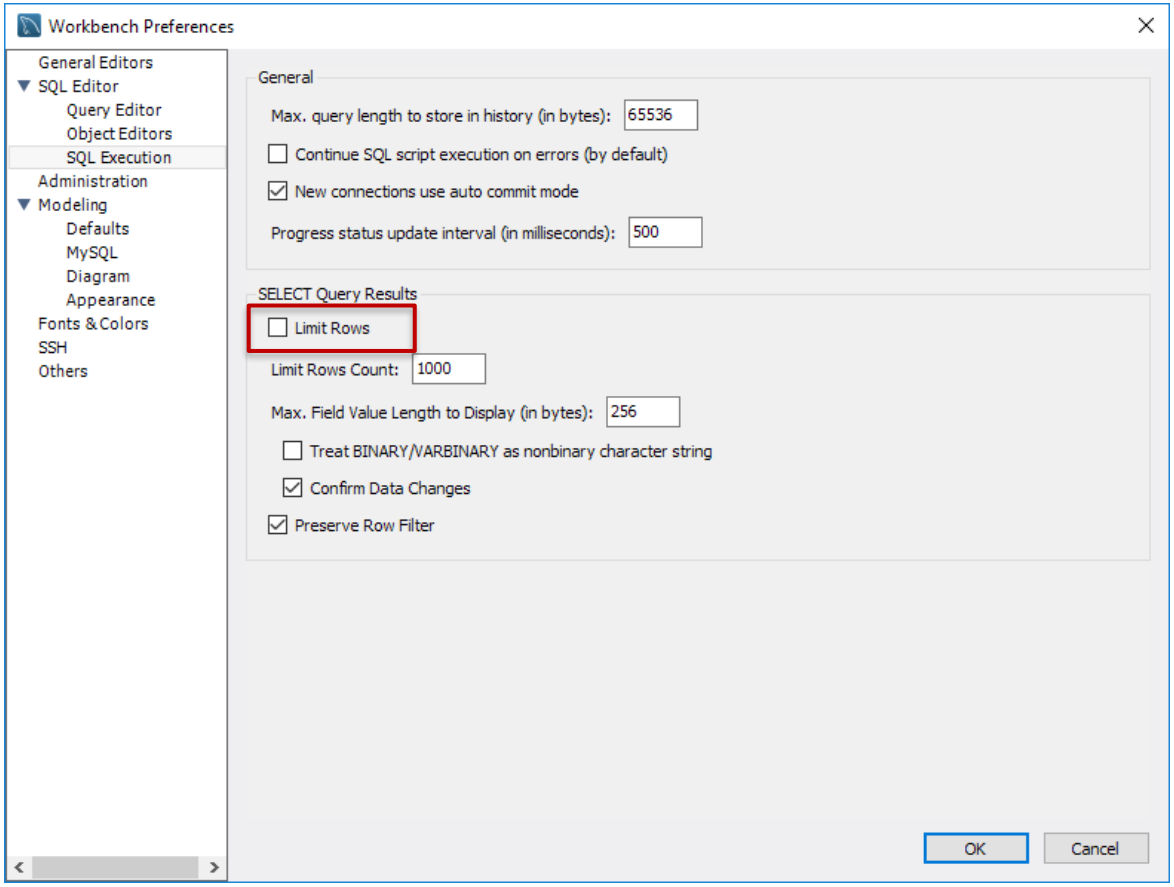
Viktig: sett World databasen som "default" db

- I MySQL Workbench må man aktivere databasen man ønsker å jobbe mot.
- Høyreklikk databasen, Set as Default Schema.
- Workbench husker valget fra gang til gang, så trenger bare gjøre dette første gangen, evt. om man vil bytte aktiv database.



Oppsett, MySQL Workbench

- Edit->Preferences i menyen.
- ”SQL Execution” raden.
- Ta vekk kryss fra ”Limit Rows”.



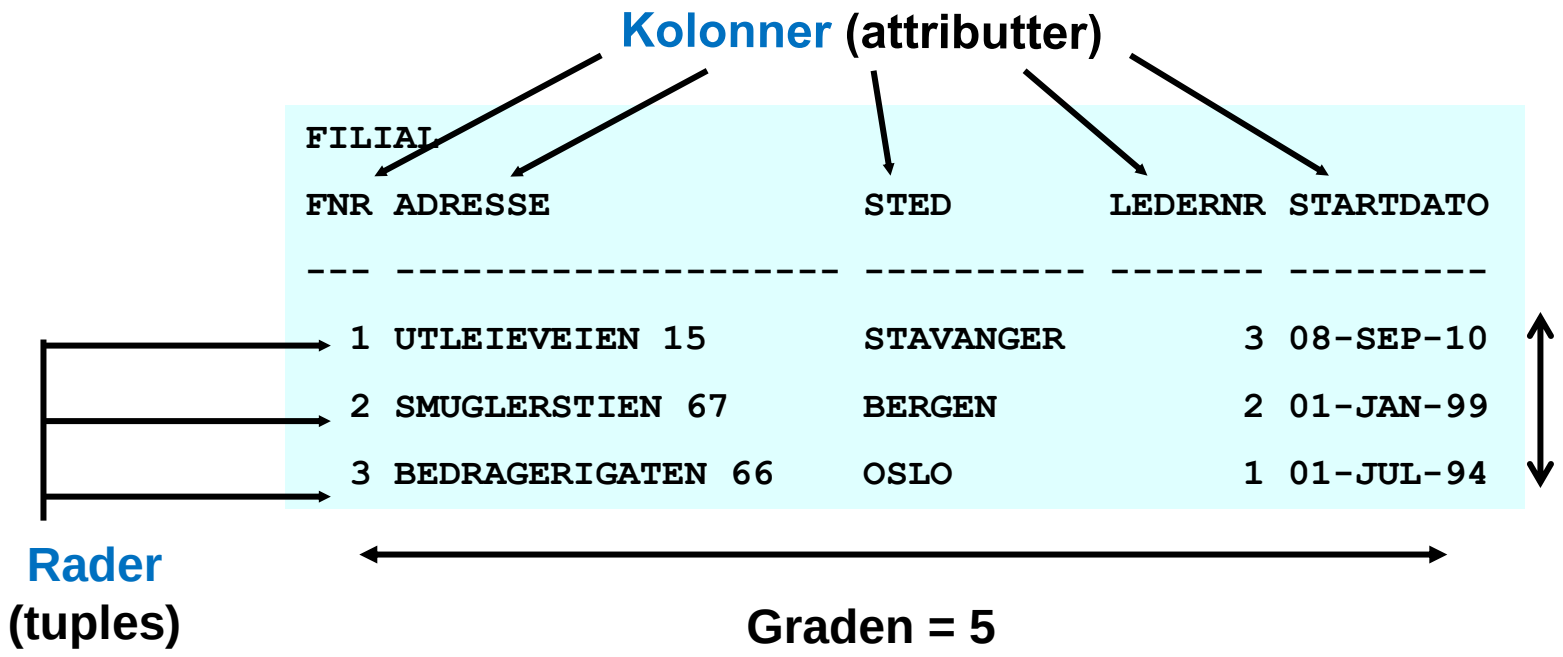
Relasjonsmodellen: Terminologi

- En relasjon er en tabell med kolonner og rader.
  - En tuppel er et annet navn for en rad i tabellen.
  - Et 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).

Rel.mod.: Terminologi – forts.



Domene

Attributt	Domenenavn	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

Datatyper

- Navn og syntaks for datatyper varierer litt fra database til database.

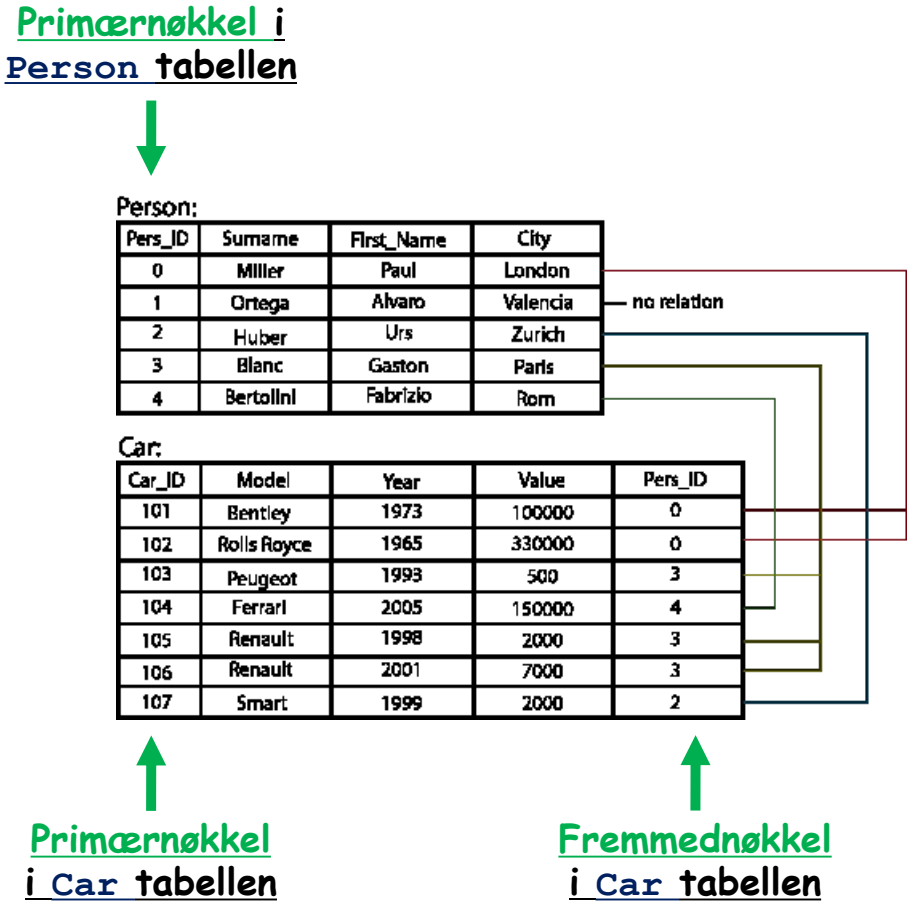
- MySQL inneholder en rekke datatyper. Blant de vanligere er: char, varchar, int, float, date og enum.

- Fullstendig oversikt (MySQL med fler) finner dere her:
  - SQL datatypes @ w3schools.com.

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.



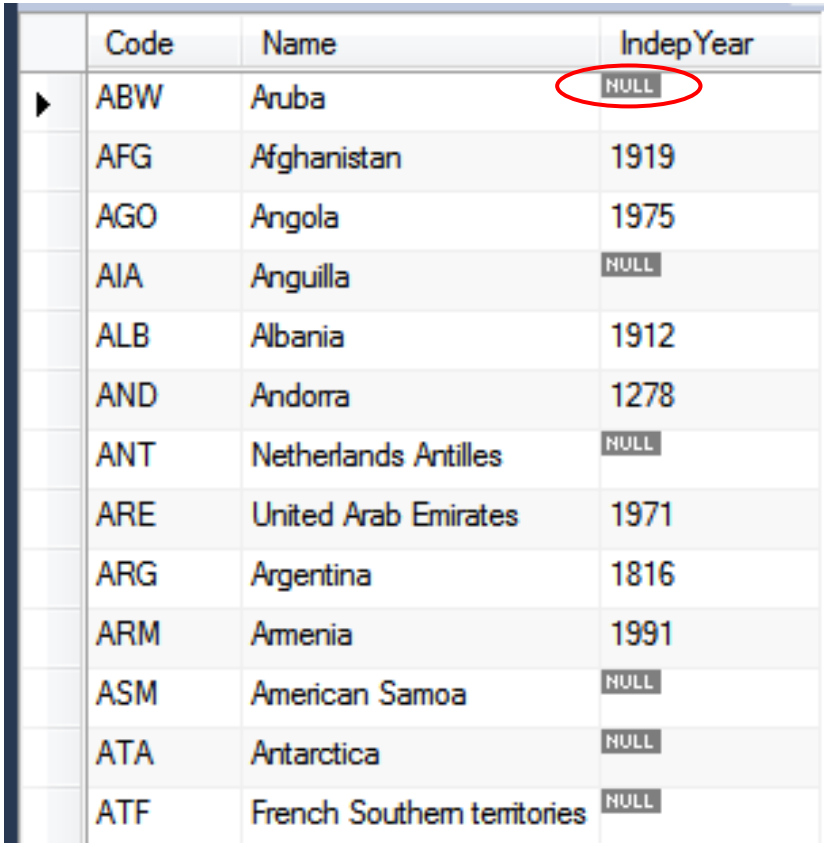
Integritet

- En primærnøkkels kolonner kan ikke inneholde NULL.
- En fremmednøkkel må ha samme verdi som primær-nøkkelen i tabellen den refererer til, eller være NULL.

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	Armenia	1991
ASM	American Samoa	NULL
ATA	Antarctica	NULL
ATF	French Southern territories	NULL

NULL kan fort spille oss et puss (1)

```
SELECT COUNT(*) AS AntLand
FROM country;
```

AntLand
239

```
SELECT COUNT(*) AS AntLand
FROM country
WHERE IndepYear > 1814
OR IndepYear <= 1814;
```

AntLand
192

NULL kan fort spille oss et puss (2)

```
SELECT COUNT(*) AS AntLand
FROM country
WHERE IndepYear > 1814
OR IndepYear <= 1814
OR IndepYear = NULL;
```

	AntLand
▶	192

```
SELECT COUNT(*) AS AntLand
FROM country
WHERE IndepYear > 1814
OR IndepYear <= 1814
OR IndepYear IS NULL;
```

	AntLand
▶	239

SQL – SELECT queries

- Hensikten med en SELECT query er å hente data fra en eller flere tabeller.
- Resultatet av en SELECT vises som en ny tabell.
- SELECT er den mest brukte SQL kommandoen.
- Syntaks/rekkefølge:

```
 select     kolonne [as navn]
 from       tabell
[where      betingelse]
[group by   grupperingsuttrykk] [having betingelse]
[order by   kolonne]
```

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:

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

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

- Logiske operatorer – setter sammen kriterier:

`and`	og
`or`	eller
`not`	ikke

GROUP BY og HAVING

- `GROUP BY` lar oss gruppere summeringsresultater 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 – forts.

```
SELECT COUNT(*), MIN(SurfaceArea),
MAX(IndepYear), AVG(LifeExpectancy), SUM(GNP)
FROM country;
```

```
SELECT Continent, COUNT(*), MIN(SurfaceArea),
MAX(IndepYear), AVG(LifeExpectancy), SUM(GNP)
FROM country
GROUP BY Continent;
```

```
SELECT Continent, COUNT(*), MIN(SurfaceArea),
MAX(IndepYear), AVG(LifeExpectancy), SUM(GNP)
FROM country
GROUP BY Continent
HAVING COUNT(*) > 20 AND MIN(SurfaceArea) > 20;
```

GROUP BY og HAVING – forts.

- `WHERE` ekskluderer rader før gruppering.
- `HAVING` ekskluderer rader etter gruppering.
- SQL utføres nemlig i følgende rekkefølge:
  - `FROM`
  - `WHERE`
  - `GROUP BY`
  - `HAVING`
  - `SELECT`
  - `ORDER BY`

(Altså ikke i kronologisk rekkefølge.)

GROUP BY og HAVING – forts.

```
SELECT Continent, COUNT(*), MIN(SurfaceArea),
MAX(IndepYear), AVG(LifeExpectancy), SUM(GNP)
FROM country
WHERE IndepYear < 1950
GROUP BY Continent
HAVING COUNT(*) > 20 AND MIN(SurfaceArea) > 20;
```

- Forskjellen på denne og forrige spørring er "WHERE IndepYear < 1950".
- Hva er forskjellen i resultatet?
- Vi «mister» Nord-Amerika fordi vi mister rader ved bruk av `WHERE`, noe som medfører at `HAVING`-betingelsen ikke lenger oppfylles.

Øving

- Før øvingen er det flott hvis du har:
  - Lest kapittel 2 i pensumboka.
  - Lest igjennom denne slideserien.
  - Kjøre spørringene, ikke bare lese dem. Gjør gjerne små endringer også, og se om du får forventet resultat.
  - Gjerne ha begynt å titte på øvingsoppgavene.
- Sjekk TimeEdit for å se når DIN øving begynner**.**
- Våre flinke studentveiledere er på plass i øvingslokalene for å hjelpe deg :-)

Om øvingen (Oslo)

- Finn en plass i et av rommene (auditoriene)
- Jeg håper vi har veiledergensere på plass…
- Jeg anbefaler fremdeles å jobbe sammen (men ingen tvang)