



Økt 1 (av 12)

DB1102 Databaser

Per Lauvås / per.lauvas@kristiania.no

Yuan Lin / yuan.lin@kristiania.no

Videoer

1. Intro til emnet
2. Om databaser
3. SQL
4. Fram mot øving 1

Pensum:

- Slides i denne presentasjonen
- Læreboka kapittel 1



Kort om meg, Per

- Lærer ved Teknologi på Høyskolen Kristiania.
- Arbeidslivet
 - 2004: Master i IT fra UiO.
 - 2004 - 2013: Utvikler (Mintra og CGI).
 - 2013 – 2018: Høyskolelektor NITH -> Westerdals -> HK.
 - 2018 - 2020: Utvikler, Universitetsforlaget.
 - 2020 -> Førstelektor, HK.
- Her på HK underviser jeg emner innen database, programmering og smidig metode.
- Er studieprogramleder for Bachelor i IT og spesialiseringsansvarlig for Programmering.



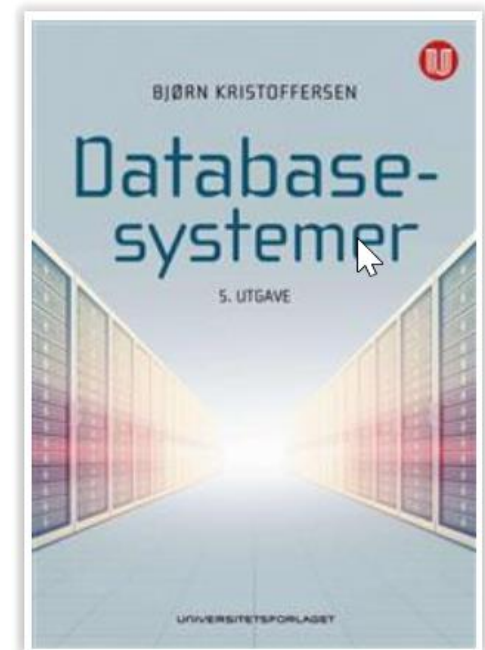
Yuan Lin

- Høyskolelektor ved Høyskolen Kristiania.
- PhD ved NTNU
- Design engineer, Nera Networks
- Design engineer Ceragon Networks
- Applikasjonanalytiker, DNB
- Systemutvikler, Sparebanken Vest
- Underviser dette semesteret:
 - Grensesnittdesign
 - Webutvikling 1
 - Database



DB1102 Databaser

- 4 timer per uke, 12 økter (totalt 48 timer lærerstyrt).
 - Pluss selvstudium: et 7,5 SP emne beregnes til 200t!
- Databaser er et modningsfag, ALLE trenger å gjøre øvinger for å bli gode!
(Kan ikke lære om databaser uten å gjøre øvingsoppgavene!)
- Lærebok:
 - "Databasesystemer", 5. utg.,
©2020, av Bjørn Kristoffersen.



[Lærebokas webside](#)

We have a plan!

DB1102 - oversikt						
Økt	Dato	Tema	Pensum (5. utg. av boka)			
1	25.aug	Introduksjon, SQL	Kap 1			
2	01.sep	Mer SQL	Kap 2			
3	08.sep	Lage og bruke tabeller	Kap 3			
4	15.sep	Likekobling (join), kryssprodukt	Kap 4			
5	22.sep	View, Delspørring	Kap 5.1-5.3			
6	29.sep	Tilbakeblikk, Arbeidskrav				
7	06.okt	ER-modellering, del 1	Over 2 økter: Kap 7 - 8.1			
8	13.okt	ER-modellering, del 2	Over 2 økter: Kap 7 - 8.1			
9	20.okt	Normalisering, UNF - 3NF	Kap 8.2.1 - 8.2.6			
10	27.okt	Normalisering, UNF - BCNF	Resten av kap 8			
11	03.nov	Diverse	Kap 9.3, 10.2.1, 10.3.2, 11.2.2			
12	10.nov	Avslutning, eksamenstips				
Eksamen 7. des kl 09.00. 24t hjemmeksamen						
Det er ett arbeidskrav i emnet som vi vil starte på i økt 6						

Flipped classroom

1. Forberedelser
 - Se igjennom videoene til økten
 - Les pensum
 - Starte på oppgavene?
2. 2t øving med studentassistenter til stede.
3. Felles samling i Zoom.

**Vi trenger et godt digitalt verktøy for
samhandling!**

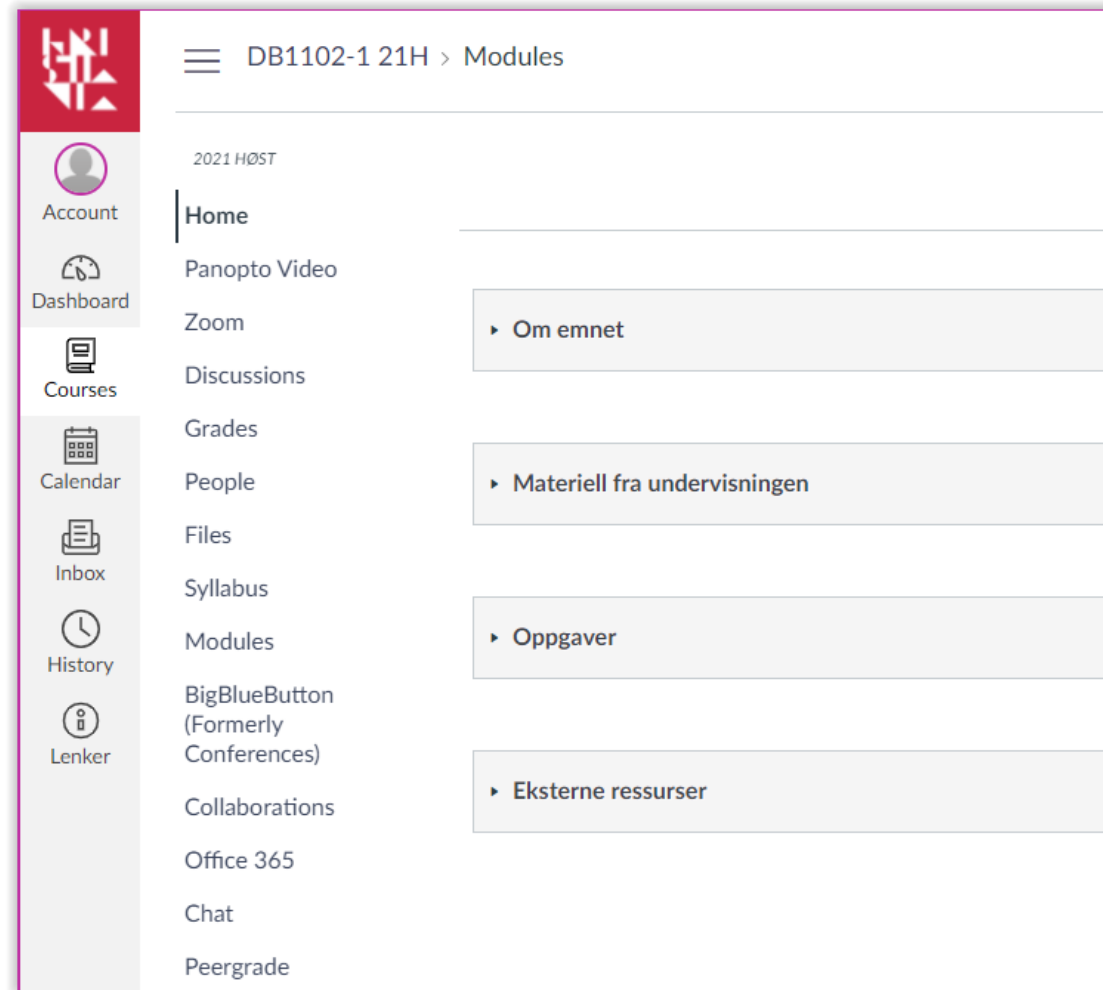


Bruk av Mattermost i DB1102

- Vi bruker Mattermost til å diskutere, be om hjelp, vise fram løsninger, lage oppgaver til hverandre, prate om ting som ikke har med fag å gjøre mm.
- MEN Canvas er fremdeles den offisielle kommunikasjonskanalen. Dvs: Hvis vi (foreleserne) har viktig informasjon å komme med, så bruker vi Canvas.

DB1102, Canvas

- Se Canvas for:
 - Emneplan DB1102
 - Slides
 - Lenke til Zoom-møter
 - Øvingsoppgaver
 - Ressurser (websites)
 - Kunngjøringer



- Og vi bruker Mattermost til å diskutere og hjelpe hverandre.

Altså...

- Før hver økt:
 - Les pensum (sjekk fremdriftsplan)
 - Se igjennom videoene i Canvas (Panopto video)
- Læringsressursene (videoer, slides og oppgaver) blir publisert i løpet av torsdagen.

Utklipp DB1102 emneplan

- Innhold

- Etter fullført Databaser 1 skal man kunne forklare hva en relasjonsdatabase er, hva den kan brukes til og hvordan den skiller seg fra andre former for persistent lagring.
- Man skal kunne modellere og strukturere data for et domene.
- Man skal kunne opprette tabeller, legge inn ulike typer data i disse, kople dem sammen og hente ut rapporter ved hjelp av SQL spørringer.
- Man skal kunne forklare og anvende prinsippene for god design (normalisering, nøkkelbruk).

DB1102 Eksamen (vurderingsform)

- Individuell hjemmeksamen 24t
 - Utgjør 100% av karakteren.
 - Ett arbeidskrav. Mer om dette i økt 6...

DA SETTER VI I GANG
MED PENSUM ! :-)

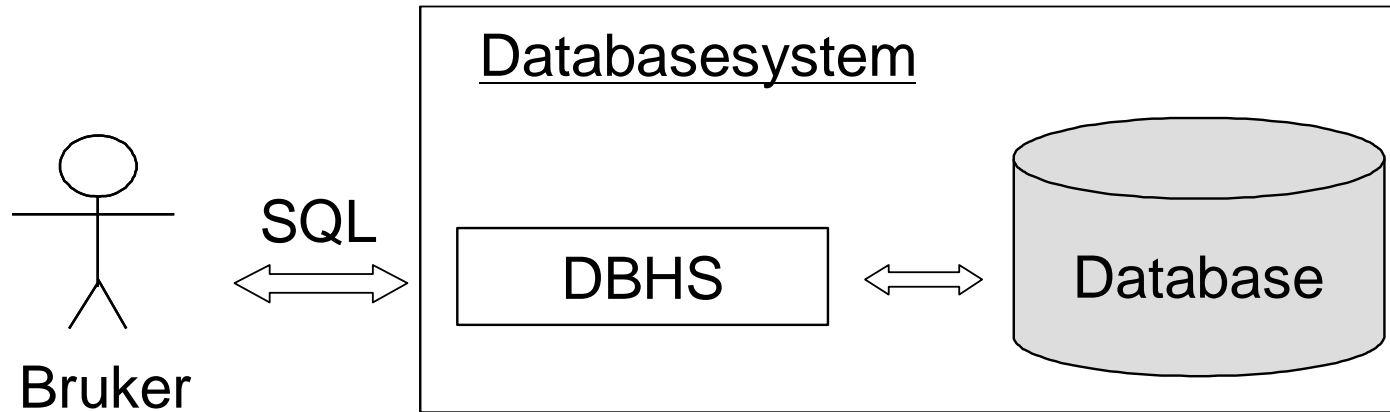
Databasebegreper

- Databasesystem = database + DBMS.
- En database er en strukturert samling av data.
- Et DBMS – DataBase Management System – (norsk: DBHS → DataBase HåndteringsSystem) er et verktøy for å lagre og gjenfinne store mengder data over lang tid, på en sikker og effektiv måte, for mange samtidige brukere.

Databasebegreper – forts.

- Et databasesystem skal ...
 - ... kunne lagre store mengder data over lang tid.
 - ... kunne lagre dataene på en sikker måte.
 - ... kunne kommunisere med andre systemer.
 - ... tilby mekanismer for å gjenfinne data effektivt.
 - ... håndtere et stort antall brukere samtidig.
 - ... håndtere kritiske situasjoner som diskkrasj og strømbrudd.
- Merk: Vi omtaler som oftest database-systemer bare som "databaser"!

Database begreper – forts.



- SQL (Structured Query Language) er et språk for å kommunisere med relasjonsdatabaser. (En vanlig form for databaser, kommer tilbake til dette etterpå.)

"Alle" bruker databaser!



Google Search

I'm Feeling Lucky

Google.no offered in: [norsk \(bokmål\)](#) [norsk \(nynorsk\)](#)

"Alle" bruker databaser!

- Ja ... "ALLE" bruker databaser! ;-)
- Et lite knippe eksempler:
 - Handel: Varer, kunder, bestillinger, leveranser ...
 - Bibliotek: Bøker, lånetakere, utlånte bøker, ...
 - Bank: Kunder, kontoer, innskudd, uttak, overføringer, ...
 - Sykehus: Pasienter, journaler, ansatte, turnus, ...
 - Kart: Eiendommer, bygninger, veier, rørsystemer, ...
 - Kino: Filmer, forestillinger, reservasjoner, ...
 - Forskning: Spørreundersøkelser, respondenter, svar, ...
- Mange systemer må være i drift 24/7 (oppetidkritiske).

Historie

- De første DBHS kom midt på 60-tallet. General electric og IBM var tidlig på banen. Det første DBHS var en nettverksdatabase.
- Deretter kom det første med kommersiell suksess og det var en hierarkisk database.
- I moderne tid er relasjonsdatabaser standarden. De bygger på teorier utviklet av Codd rundt 1969/1970.
- Codd, E. F.: "A Relational Model of Data for Large Shared Data Banks", in Communications of the ACM, 1970.

Historie – forts.

- I nyere tid har vi også fått andre typer databaser som objektdatabaser og grafdatabaser. En ny variant som har vind i seilene: dokumentdatabaser.
- Databaser som ikke er relasjonsdatabaser kalles gjerne NoSQL databaser («Non SQL» eller «Not Only SQL»).
- Vi konsentrerer oss om relasjonsdatabaser.
 - Eksempler: PostgreSQL, SQL Server, Oracle, [MySQL](#).
- MySQL er, i følge dem selv, den mest brukte SQL databasen.

Relasjonsdatabaser

- En relasjonsdatabase ("RDBMS") er bygget opp av relasjoner med data.
 - En relasjon er en tabell med kolonner og rader.
 - Vi bruker heller begrepet «tabell» enn «relasjon».
 - Tabellen er altså fundamentet i relasjonsdatabasen.
- Tabellene, og dataene i dem, henger sammen gjennom bruk av nøkler (eng: keys) i de forskjellige tabellene.
 - Sammenhengen mellom tabeller kalles gjerne relasjon, noe som gjør dette ganske så forvirrende...

Relasjonsdatabaser – forts.

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

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

no relation

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

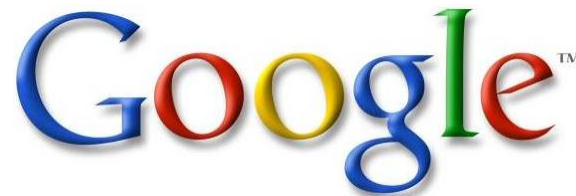
Fremmednøkkel
i Car tabellen

MySQL – database for DB1102

- MySQL er en relasjonsdatabase.
 - Open-source, under GNU lisensen.
- MySQL eies nå av Oracle.
 - Opprinnelig eid av MySQL AB i Sverige.
 - Oppkalt etter utviklerens (Michael Widenius) datter: My
 - (Tilleggsinfo for prog'erne: Kodet i C/C++.)
- Kilde: Wikipedia

MySQL – forts.

- MySQL benyttes bl.a. av:



WIKIPEDIA
The Free Encyclopedia

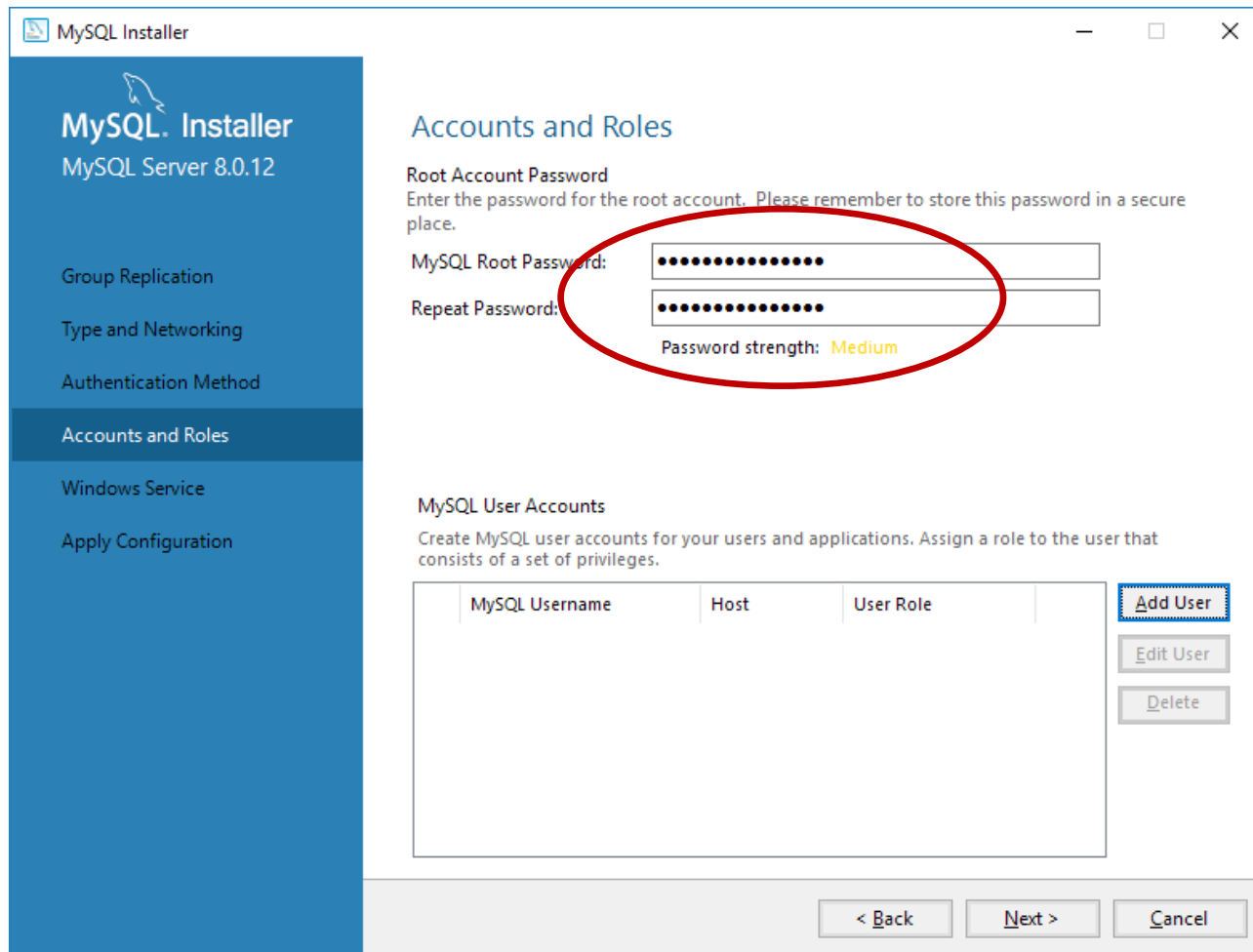


MySQL – forts.

- MySQL kan bl.a. kjøres på:
 - Mac OS X
 - Windows
 - Linux
- Installasjon er avhengig av OS...
- Vi har rykende ferske installasjonsveiledninger som ligger i Canvas (under Eksterne ressurser)



Husk dette passordet!



MySQL Installer
MySQL Server 8.0.12

- Group Replication
- Type and Networking
- Authentication Method
- Accounts and Roles**
- Windows Service
- Apply Configuration

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

Repeat Password:

Password strength: **Medium**

MySQL User Accounts
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

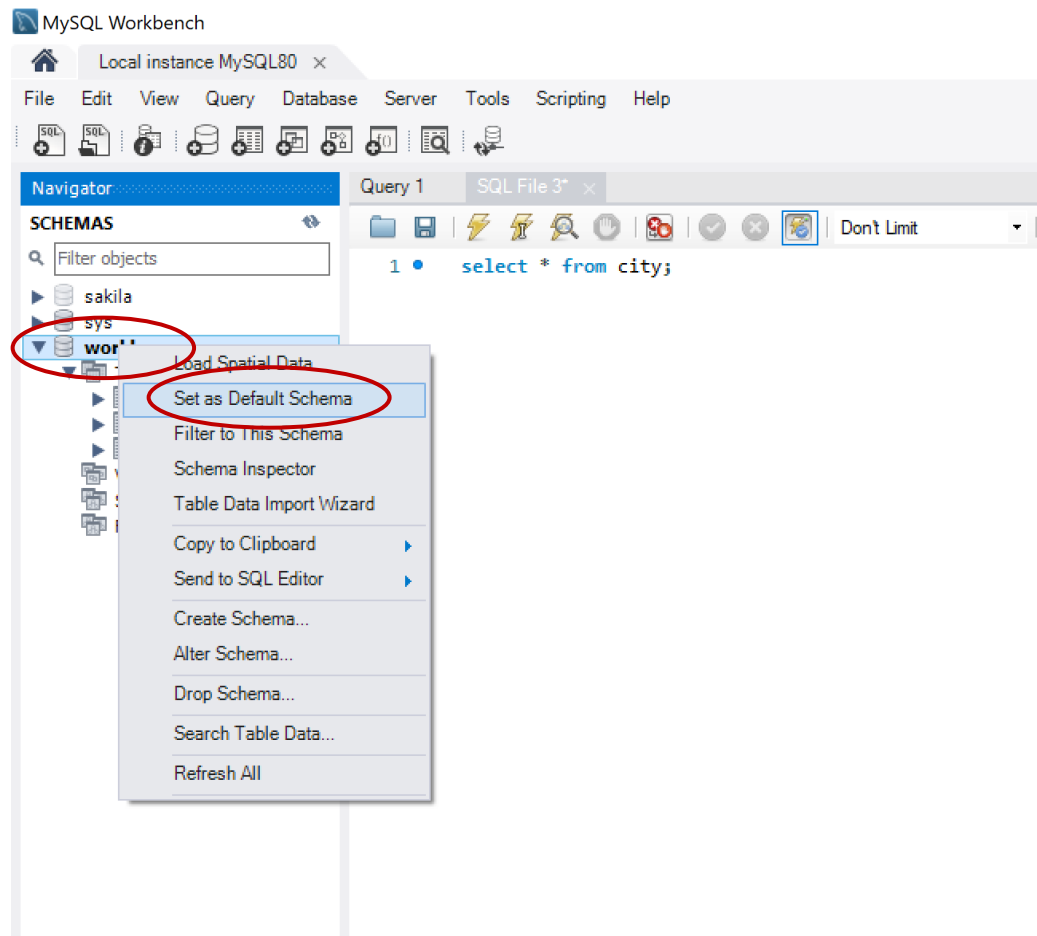
MySQL Username	Host	User Role
----------------	------	-----------

[Add User](#)
[Edit User](#)
[Delete](#)

< Back Next > Cancel

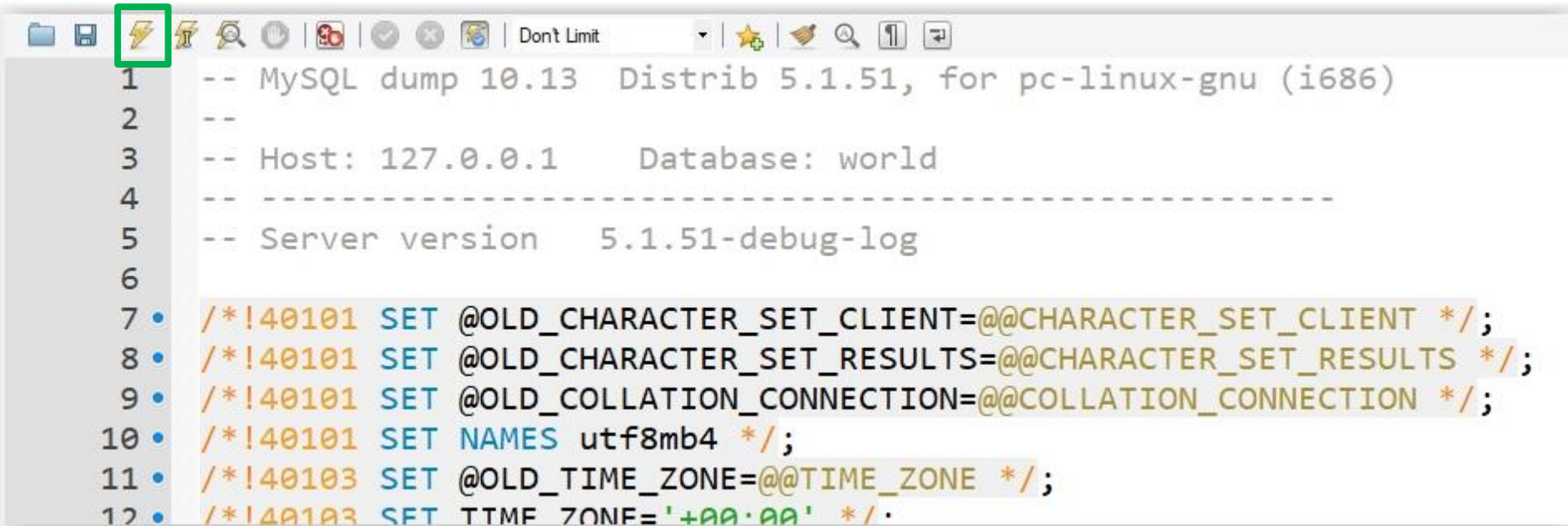
Viktig: sett World databasen som "default" db

- 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.



Problemer med installasjonen?

- Sjekk «Eksterne ressurser» i Canvas.
- Hvis du glemte å installere «World»-databasen, så trenger du ikke å installere alt på nytt. Da kan du bruke world-schema (ligger også i Canvas) til å laste inn databasen. Kopier all teksten i filen, lim det inn i Workbench (Query-vinduet). Klikk «lyn»-knappen (Execute).



```
1  -- MySQL dump 10.13  Distrib 5.1.51, for pc-linux-gnu (i686)
2  --
3  -- Host: 127.0.0.1    Database: world
4  -- -----
5  -- Server version    5.1.51-debug-log
6
7  • /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8  • /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9  • /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 • /*!40101 SET NAMES utf8mb4 */;
11 • /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 • /*!40103 SET TIME_ZONE='+00:00' */;
```

Håndtering av data

- Vi bruker databasespråket SQL.
 - SQL reguleres av standarder fastsatt av ISO.
- Støttes i praksis av alle RDBMS (inklusive MySQL databasen, som vi benytter).
- Merk: De fleste RDBMS har noen egne tilleggstilpasninger av SQL. ("Dialekter")

Datamaskiner og formelle språk

- SQL (i motsetning til norsk eller fransk) er et formelt språk: Det er definert presise regler som avgjør om en setning er lovlig eller ikke.
- Datamaskiner har ikke evnen til å bruke sunn fornuft. Den minste skrivefeil fører til et uventet resultat eller en feilmelding.
 - Eksempel: SQL forstår ikke at du mener «Etternavn» hvis du skriver «Etternavnet».
 - Men, merk: SQL er som regel ikke "case sensitivt" – m.a.o. det bryr seg ikke om stor eller liten bokstav. Les mer [her](#).
 - Det er vanlig å skrive reserverte ord med store bokstaver, eks:
`SELECT Name FROM city;`

SQL spørringer – format

- HVA skal hentes: **SELECT** bla-bla-bla
- FRA hvilken tabell: **FROM** bla-bla-bla
- BETINGELSER som begrenser svaret: **WHERE** bla-bla-bla
- SORTERING: **ORDER BY** bla-bla-bla

SQL spørringer – forts.

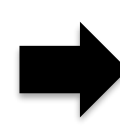
Merk: De følgende eksemplene er gjort mot MySQL eksempeldatabasen "world", som er en av databasene som følger med MySQL installasjonen.

Prøv selv, med MySQL Workbench! :-)

Noen spørringer

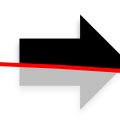
- Når vi vil ha alle dataene i en tabell:

```
SELECT *  
FROM city;
```



- Når vi kun er interessert i noen av kolonnene i en tabell:

```
SELECT Name, Population  
FROM city;
```



Merk: I følge SQL standarden skal alle spørringer ha semikolon til slutt. MySQL Workbench bryr seg ikke om dette.

Noen spørringer – forts.

- Når vi kun er interessert i noen av radene i en tabell:

```
SELECT *  
FROM city  
WHERE CountryCode = 'NOR';
```



- En betingelse (WHERE ...) er et uttrykk som per rad er sant eller galt.
 - NB: Ønsker vi flere spørrekriterier grupperes de med AND eller OR, ikke med komma!

Noen spørringer – forts.

- Sortering av resultat:

```
SELECT *  
FROM city  
ORDER BY CountryCode ASC, Population DESC;
```

- Sorteringskriterier:
 - Stigende rekkefølge: ASC (evt. ingenting)
 - Synkende rekkefølge: DESC
- Merk: ORDER BY kan også benytte kolonnennummer fra select-delen, ikke navn (ORDER BY 2 DESC e.l.). MEN: Bruk kolonnenavnet.

Spørreresultat – eksempel

- 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

- 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

Noen erfaringer fra demonstrasjonen

- Ja, ; har noe å si når vi har flere SQL-er i editoren vår. Det viser hvor en SQL slutter. Avslutt derfor en SQL med en ;
- Rekkefølge er viktig når vi skriver SQL-en vår
 - SELECT
 - FROM
 - WHERE
 - ORDER BY

Litt om øvingstimene

- Den viktigste delen av DB1102 ift eksamen og din første jobb etter studiene.
- I din første jobb, så jobber du med stor sannsynlighet sammen med andre. Øvingstimene gir deg god trening...
- Erfaringsmessig vil man ha godt av å jobbe i gruppe, men dere velger selv...

Gode råd fra veilederne

- Vær aktiv.
- Aldri «copy/paste».
- Ta oppgaven ett skritt lenger.
- Jobbe sammen med andre?
- Skaffe deg gode vaner.
- Forbered deg til hver økt.
- Bruk forumet flittig. (Kan nå oversettes til: Bruk *Mattermost* flittig.)

Første øving, onsdag 25. aug

- Du starter kl 08.15 eller 10.15 (basert på hvilken klasse du går i).

Ons. 25.08.2021	
08:15 21BIT, 21BIT DB1102, Databaser 21WESHO1 21WFMHO1 21WIDHO1	10:00
10:15 21BIT, 21BIT DB1102, Databaser 21BCSHO1 21WPGHO1 T24 TAR-200	12:00
12:00 Møte_SEIT, Møte_SEIT 0 Ansatte SEIT 2,00	14:00
10:15 DB1102, Databaser 21WESHB1 21WFMHB1 BT-510 UNDERSVIS	12:00
13:15 21BIT, 21BIT DB1102, Databaser DB1102, Databaser 21BCSHO1 21WESHB1	15:00

Og...

- Øvingsoppgavene i dag er de viktigste vi har i løpet av høsten:
 - Blir du forsinket med å installere en fungerende MySQL, kommer du skjevt ut!
- Dagens oppgave 5 (og tilsvarende i de første ukene) inkluderer bruk av Mattermost. Jeg håper å se mange gode oppgaver!

Hva har vi vært gjennom i dag?

- Databasesystem.
- Relasjonsdatabase.
- Enkel SQL.

Leseplan de neste ukene (ligger også i Canvas)

DB1102 - oversikt						
Økt	Dato	Tema	Pensum (5. utg. av boka)			
1	25.aug	Introduksjon, SQL	Kap 1			
2	01.sep	Mer SQL	Kap 2			
3	08.sep	Lage og bruke tabeller	Kap 3			
4	15.sep	Likekobling (join), kryssprodukt	Kap 4			
5	22.sep	View, Delspørring	Kap 5.1-5.3			
6	29.sep	Tilbakeblikk, Arbeidskrav				
7	06.okt	ER-modellering, del 1	Over 2 økter: Kap 7 - 8.1			
8	13.okt	ER-modellering, del 2	Over 2 økter: Kap 7 - 8.1			
9	20.okt	Normalisering, UNF - 3NF	Kap 8.2.1 - 8.2.6			
10	27.okt	Normalisering, UNF - BCNF	Resten av kap 8			
11	03.nov	Diverse	Kap 9.3, 10.2.1, 10.3.2, 11.2.2			
12	10.nov	Avslutning, eksamenstips				
Eksamen 7. des kl 09.00. 24t hjemmeksamen						
Det er ett arbeidskrav i emnet som vi vil starte på i økt 6						

Populære nettsteder innen SQL

- [sqlbolt](#)
- [w3school](#)

Lykke til med oppstarten!

Vi sees i øving (fysisk) og i felles samling (Zoom).