# Økt 3 – *Flipped versjon*

**DB1100 Databaser**

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

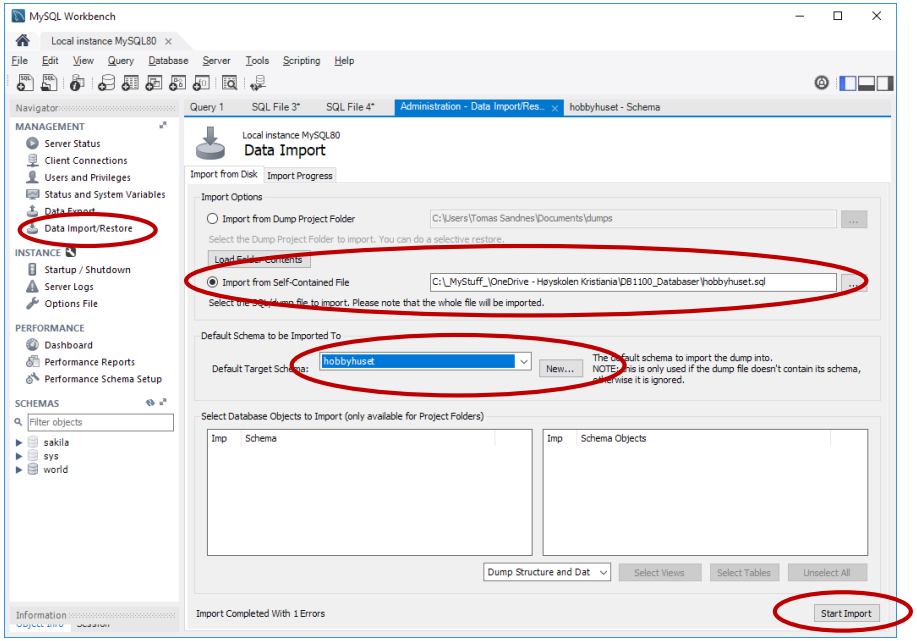
# FØR øving – HUSK!

- Les relevant kapittel i pensumboka!
  - Oversikt over kapitler finnes i "DB1100_Struktur_Innhold.jpg" på Canvas.
- Les dagens forelesningsslides.
  - Og gjerne: Se igjennom nye videoer i spillelisten.
- Om du har tid igjen før øving: Begynn på oppgavene på egenhånd.
  - Eller hva med å gjøre de i … gruppe? :-D
- Nå er du/dere klare!
  - Du vil ha lært mye.
  - Du vil være forberedt til eksamen!
  - Du vil herje i Kahooten!!!

# Oppfølging student: Legge inn Hobbyhuset db

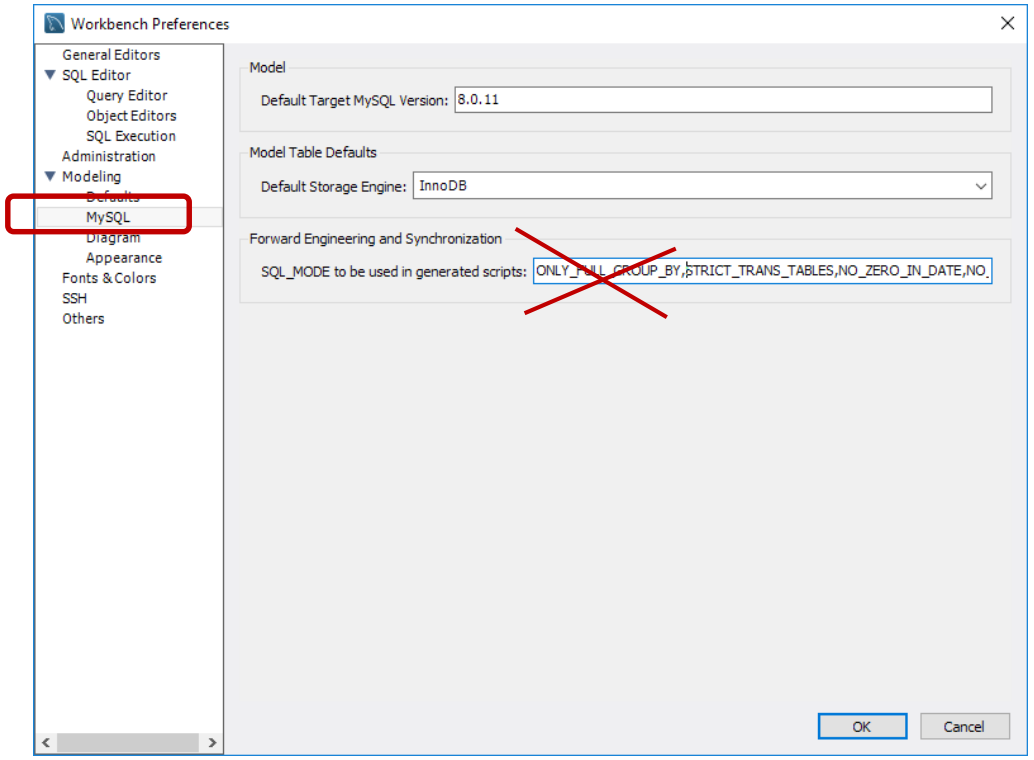
**Bruk denne linken, enklere enn den forrige.**

- For de som vil leke seg med oppgaven på Canvas, trenger man Hobbyhuset d
  - Gå til: dbsys.info/Databasesystemer/1_Datasett/sqlskript.html (nettsiden til læreboka). Høyreklikk [Tabeller + Eksempeldata], "Save link as…", lagre på din maskin.
  - I MySQL Workbench, klikk "Data Import/Restore".
  - Velg "Import from Self-Contained File", og finn fila du lastet ned.
  - Klikk [New…] knappen for Default Target Schema, skriv inn "hobbyhuset".
  - Trykk [Start Import] knapp.



# Oppfølging student: GROUP BY og OS X

- Noen har hatt litt problemer med visning av resultater i GROUP BY uttrykk på OS X.
- Kan det være at dere også har lagt inn world databasen manuelt?
- Prøv å fjerne: "ONLY_FULL_GROUP_BY," i preferences, MySQL raden.



# Datatyper i MySQL

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

- Blant MySQLs vanligere datatyper er:
  - char og varchar
  - [tiny/small/mediu
  - float
  - enum
  - date
  - timestamp

*"Selvdefinert datatype"?*

← desimaltall

← spesiell, selvdefinert datatype

← datoformat, på formen: 'YYYY-MM-DD'

← auto-tidspunkt (v/insert el. update)

- For utfyllende oversikt over datatyper i MySQL, se f.eks.:
  - http://www.w3schools.com/sql/sql_datatypes.asp

# SQL DDL m.m: endring tabeller

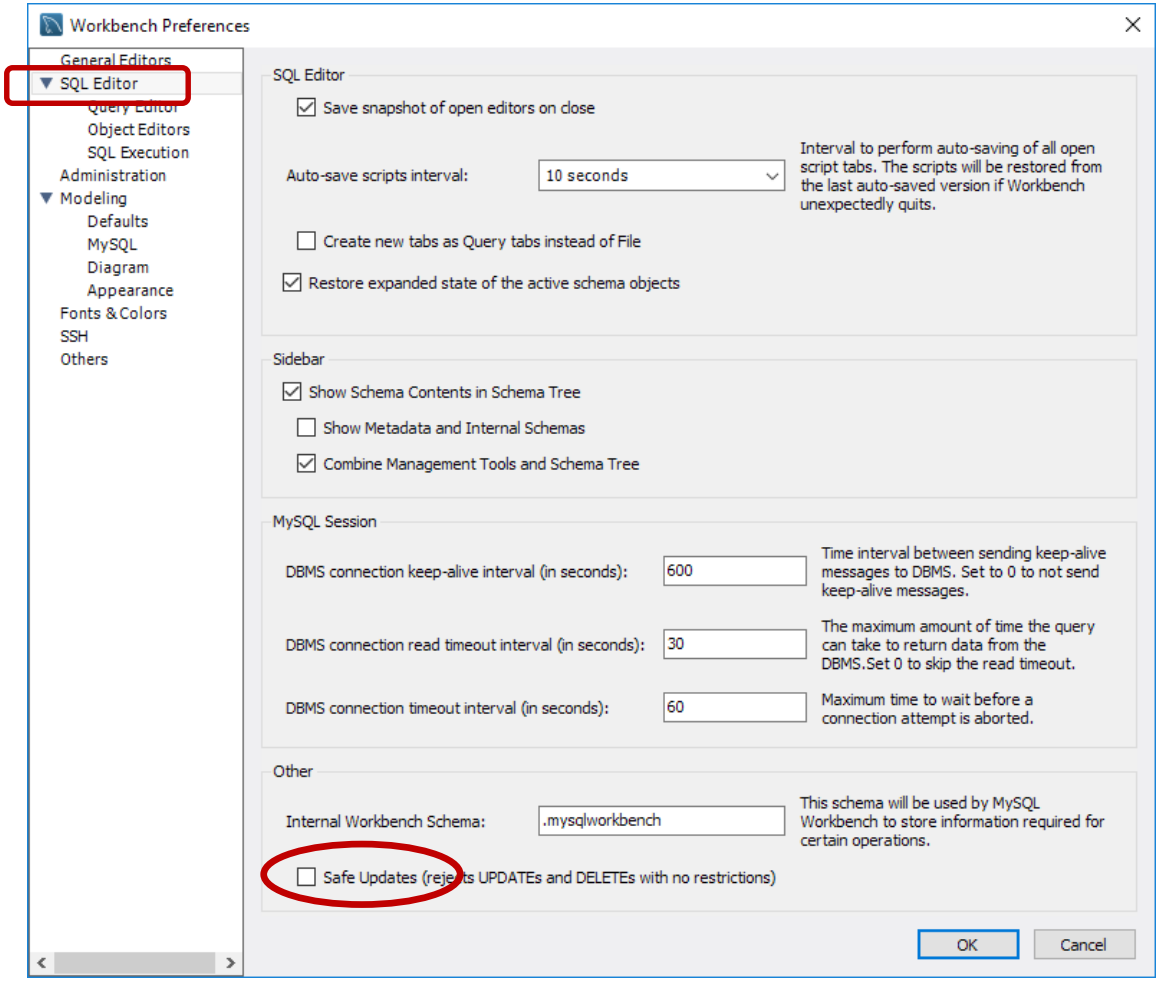
- DDL = Data Definition Language. Del av SQL. Vi har følgende SQL kommandoer for å modifisere tabeller:
  - CREATE TABLE
  - ALTER TABLE
  - DROP TABLE

- Og følgende SQL kommandoer for å modifisere innhold: (NB – ”vanlig” SQL, ikke del av DDL)
  - INSERT INTO
  - UPDATE
  - DELETE FROM

# En innstilling til i MySQL Workbench

VIKTIG!

- Edit -> Preferences...
- FJERN kryss i "Safe Updates …" feltet.
- Hindrer oss fra å gjennomføre enkelte av dagens oppgaver!
- (Mulig du må scrolle ned for å se denne innstillingen i ditt preferences vindu.)



# SQL – opprette tabell

- Skal vi lage en ny tabell, gjøres det med følgende syntaks: (et utvalg av mulighetene)

```
CREATE TABLE tabellnavn
(
kolonnenavn datatype [UNIQUE|NOT NULL|DEFAULT|...],
kolonnenavn datatype [UNIQUE|NOT NULL|DEFAULT|...],
...,
[PRIMARY KEY (kolonnenavn)],
[FOREIGN KEY (kol.navn) REFERENCES tabell_navn (kol.navn)]
);
```

# SQL – opprette tabell (forts.)

- Eksempel #2: en tabell med fremmed-nøkkel til countryIsh tabellen:

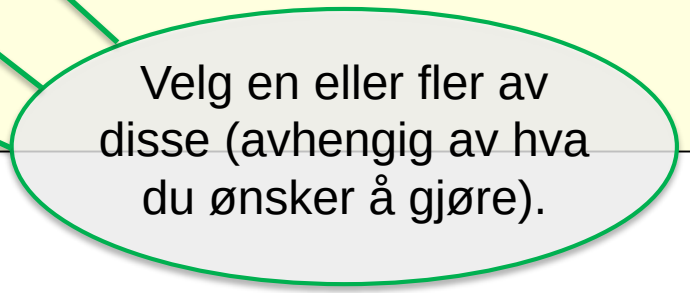
```
CREATE TABLE cityIsh
  (
     id          INT(11) NOT NULL AUTO_INCREMENT,
     name        CHAR(35) NOT NULL DEFAULT '',
     countryCode CHAR(3) NOT NULL,
     district    CHAR(20) NOT NULL DEFAULT '',
     population  INT(11) NOT NULL DEFAULT 0,
     PRIMARY KEY (id),
     CONSTRAINT fk_city_countryIsh FOREIGN KEY
       (countryCode) REFERENCES CountryIsh(Code)
  )
```

  - (AUTO_INCREMENT betyr, som alt nevnt, autogenerert løpenummer.)

# SQL – endre tabell

- Eksisterende tabeller kan endres med følgende syntaks: (MySQL spesifikt)

```
ALTER TABLE tabellnavn
ADD kolonnenavn datatype,
CHANGE kolonnenavn_nå kolonnenavn_ny datatype_ny,
DROP COLUMN kolonnenavn,
...
```




- Eksempel:

```
ALTER TABLE person
ADD foedselsdato date;
```

# SQL – slette tabell

- Å slette en tabell er kort og greit:

```
DROP TABLE tabellnavn;
```

  - NB: Men vær sikker på at du sletter rett tabell! Den blir borte for evig og alltid. ;-)

- Eksempel:

```
DROP TABLE epost;
```

# SQL – legge til innhold i tabell

- Å legge til rader i en tabell kan gjøres på to måter.
  - Med eller uten spesifisering av kolonnene:

```
INSERT INTO tabellnavn
VALUES (value1, value2, value3, ..., valueN);
```

eller

```
INSERT INTO tabellnavn (kolonnenavn1, kolonnenavn2, ...)
VALUES (value1, value2, ...),
       (value1, value2, ...),
       ...
```

# Innsettingsregler

- Dersom vi setter inn data i alle kolonner, trenger vi ikke ramse opp kolonnene.

- Skal vi sette inn data i bare noen kolonner, må vi navngi kolonnene som skal ha data.
  - For kolonnenavn som sløyfes fra listen blir det satt inn `null` (eller default verdier).

- Tekstfelt (som `varchar`) skal stå i fnutter!

- Husk: innholdet i primærnøkkelceller må være unikt!

# INSERT INTO – eksempel

```
INSERT INTO countryIsh (Code, Name, Continent)
VALUES ('WWW', 'Wlandet', 'Asia');
```

```
INSERT INTO cityIsh (name, countryCode, population)
VALUES ('Byen min', 'WWW', 12345);
```

- Legg merke til at vi ikke legger inn infomasjon i alle kolonner, selv om de ikke tillatter nullmeker.
  - Vi kan gjøre det slik fordi de får satt en DEFAULT-verdi.

- Er det tilfeldig at det legges inn informasjon i tabellen `countryIsh` først?

# SQL – oppdatere innhold i tabell

- Oppdatering av innhold i en tabell kan gjøres for alle rader, eller med en where-clause for et utvalg rader:

```
UPDATE tabellnavn
SET kolonnenavn1 = value1, kolonnenavn2 = value2, ...
[WHERE ...]
```

- Eksempel:

```
UPDATE countryIsh
SET Region = 'Wregion', Population = 74
WHERE Code = 'WWW';
```

# SQL – fjerne innhold i tabell

- Fjerning av innhold i en tabell kan gjøres for alle rader, eller med en where-clause, for enkelte rader: (som for update)

```
DELETE FROM tabellnavn
[WHERE ...]
```

- Eksempel:

```
DELETE FROM countryIsh
WHERE Code = 'WWW';
```

- NB: Som en sikkerhetsfeature har MySQL Workbench som default en sperre mot delete from uten where-clause som inneholder key referanse. (Dette kan endres i Preferences.)

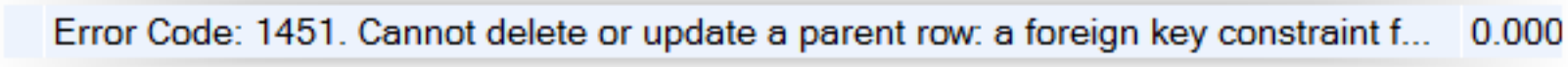
# Lagre/angre

- Når vi endrer data og/eller tabeller i SQL, er det to måter dette kan håndteres på.

- Endringene kan gjøres gjeldende fortløpende.
  - Dette kalles auto-commit, og er default settingen i MySQL Workbench.

- Endringene kan gjøres gjeldende først når vi ber om det (ergo kan vi gjøre fler samlet, og de kan angres).
  - Vi styrer dette med SQL-kommandoene `START TRANSACTION`, `COMMIT` og `ROLLBACK`.
  - Finner du knappen for å toggle autocommit i MySQL Workbench?

# What? DELETE fungerte ikke!?!




- Vi får ikke til å slette fra `countryIsh`.
  - Vi får ikke lov fordi byen vi la inn har en fremmednøkkel til dette landet.

- Vi kan fikse dette ved å slette byen først. Men la oss forsøke noe annet:
  - Vi vil legge inn en `DELETE CASCADE` (pensumboka kap. 3.2.7).
  - Da må vi fjerne FK, og legge den inn på nytt (neste side).

# CASCADE DELETE, eksempel

- Først fjerne eksisterende FK:

```
ALTER TABLE cityIsh
DROP FOREIGN KEY fk_city_countryIsh;
```

- Deretter opprette den på nytt med `DELETE CASCADE`:

```
ALTER TABLE cityIsh
ADD CONSTRAINT fk_city_countryIsh
FOREIGN KEY (countryCode) REFERENCES
CountryIsh(Code)ON DELETE CASCADE;
```

- Forsøker så å slette på nytt:

```
DELETE FROM countryIsh
WHERE Code = 'WWW';
```

# Gjøre oppdateringer via GUI i Workbench?

- Det vi har sett på i dag kan løses "enklere" i MySQL Workbench:
  - Det finnes GUI-valg (brukergrensesnitt) som lar oss gjøre de samme tingene uten å skrive SQL statements.

- MEN:
  - Hva når vi skal opprette en større database? F.eks. world fra "world_schema.sql" – skal vi opprette ca. 5000 rader for hånd?
  - Hva når vi skal distribuere en databaseløsning sammen med resten av softwaren vår, slik at den opprettes på sluttbrukerens maskin. Skal vi ringe på døra for å gjøre det gjennom GUI?

- SQL statements er ikke alltid den enkleste måten å lage eller endre tabeller på. Men av og til er det den eneste, praktiske måten!

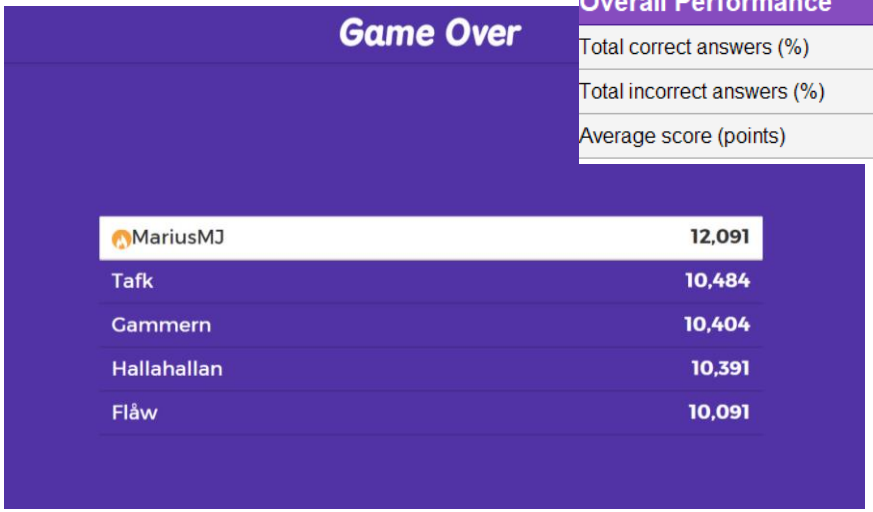
# Resultat, forrige Kahoot

- Selv med to vriene spørsmål "knuste" dere fjorårets studenter! ;-)
  - De klarte oppunder 61 % riktig svar.

- Dere: 64 % riktig svar!
  - Gi dere selv noen velfortjente klapp på skulderen! :-D

| DB1100 - uke 2 | |
|---|---|
| Played on | 30 Aug 2018 |
| Hosted by | Tomas_Sandnes |
| Played with | 169 players |
| Played | 10 of 10 questions |

| Overall Performance | |
|---|---|
| Total correct answers (%) | 64,06% |
| Total incorrect answers (%) | 35,94% |
| Average score (points) | 5745,53 points |



# NY Kahoot - hva husker vi fra i dag?

- https://kahoot.it/#/
  - Eller gjerne Kahoot appen for mobil.

- **Skal dere slå fjorårets studenter igjen!? :-D**

- Deres score: 63,2 %

| Overall Performance | |
|---|---|
| Total correct answers (%) | 63,20% |
| Total incorrect answers (%) | 36,80% |
| Average score (points) | 6189,37 points |

# NB – neste gang!

- Neste gang: Kapittel 4 – Spørringer mot flere tabeller.
- Neste økt ikke før torsdag om 2 uker!
- Dvs: God mulighet til å *repetere fra økt 1-3* neste torsdag! :-)