Økt 11 (av 12)

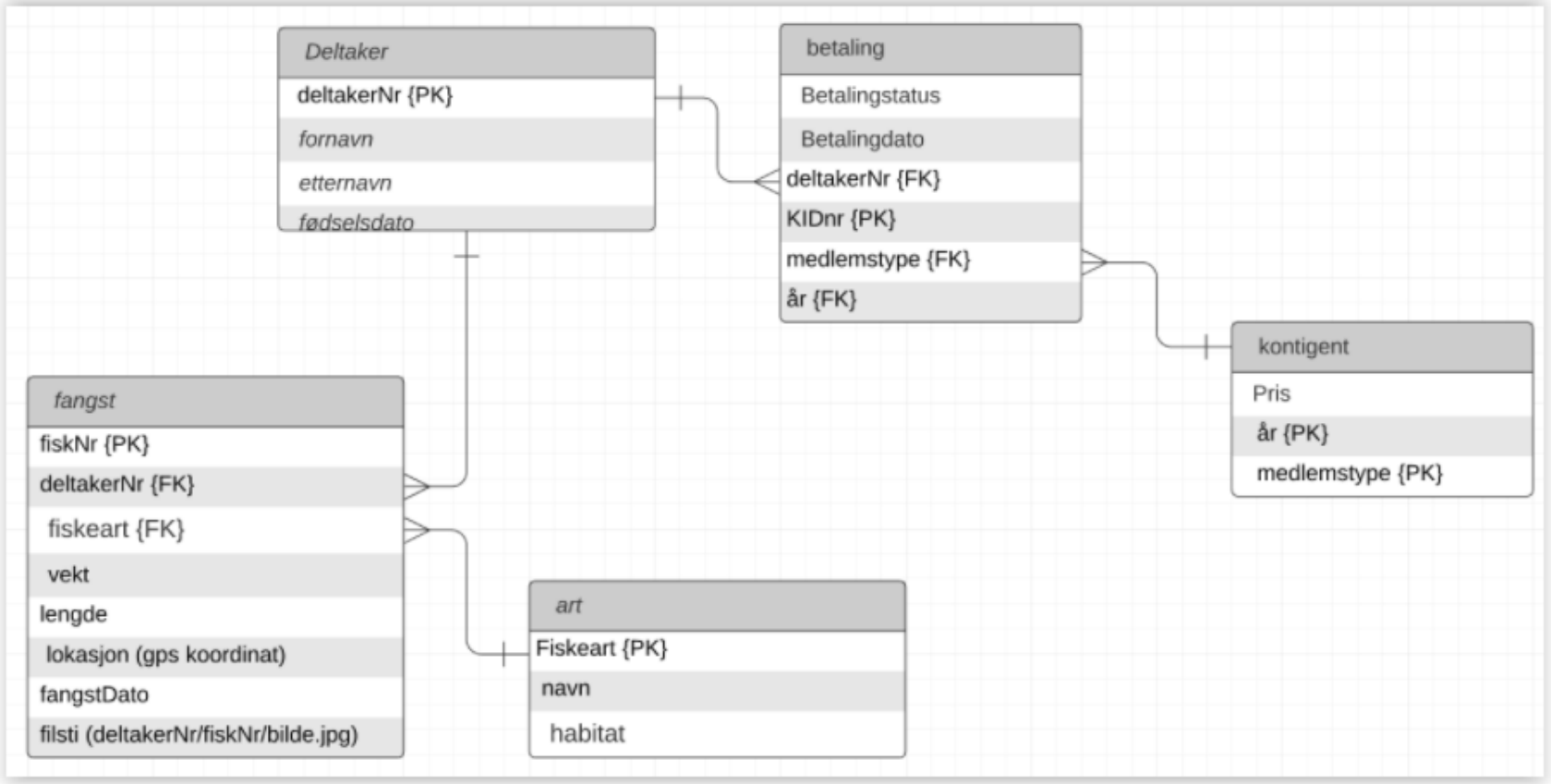
DB1102 Databaser

Per Lauvås / per.lauvas@kristiania.no

Yuan Lin / yuan.lin@kristiania.no

Gårsdagens Kahoot - pallen

Fiskekonkurransen



Dagens temaer

Dagens pensum: Læreboka 9.3, 10.2.1, 10.3.2, 11.2.2
(NB: En del av dagens pensum – personvern & GDPR – står ikke i læreboka!)

- Idag, diverse småtemaer:
  - SQL: triggere & indekser
  - Transaksjoner
  - ACID-egenskapene
  - Databaseadministrasjon
  - Lover og regler i Norge: Lagring og bruk av personopplysninger

Triggere

- I en del SQL varianter, inklusive MySQL, kan man benytte triggere.
  - Triggere kan settes til å fyre av på insert, update og/eller delete statements.

- Triggere kan for eksempel benyttes for å:
  - Logge hendelser.
  - Automatisk legge til en timestamp.
  - Rydde opp på relevante steder (f.eks ved en `delete`).
  - Si ifra når noe spesielt skjer: noe er tomt, noe er over en viss grense, osv.
- Hvorfor ikke la en tidligere student vise et eksempel på dette gjennom denne videoen (ligger i spillelisten til Olav Sundfør). Da får du i samme slengen lært om DELIMETER☺

Indekser

- Hvis du ønsker å øke ytelsen på spørringer i databasen din, er indekser første løsning du bør se nærmere på:
  - ÉN indeks kan gjøre spørringer HUNDREVIS av ganger raskere!

- Eksempel: Kjør følgende spørring, og noter deg tiden den bruker:

```
SELECT * FROM city WHERE name IN (SELECT name FROM city);
```

- Dette gikk sannsynligvis ganske fort?
  - Lag flere subqueries inne i denne igjen, slik at spørringen tar noen sekunder. Eks.:

```
SELECT * FROM city WHERE name IN
  (SELECT name FROM city WHERE name IN
    (SELECT name FROM city WHERE name IN
     (SELECT name FROM city WHERE name IN
      (SELECT name FROM city))));
```

  - Hensikt: Vi ønsker å ende opp med en spørring med kjøretid på flere sekunder.
  - (Legg til ennå flere subqueries ved behov.)

Indekser – forts.

- Når du har justert spørringen slik at den tar flere sekunder å kjøre, lag følgende Index:

```
CREATE INDEX city_name_index
ON city(name);
```

- Kjør spørringen på nytt, se hvor lang tid den benytter nå.
  - Imponerenede ytelsesøkning, eller hva!? :-D

- Hos meg gikk tiden fra ca. 4 sekunder (før index) til 0.03 sek. (etter index).
  - Det er over 100 ganger raskere bare på å legge til en index!

- Om du senere vil slette indeksen igjen:

```
DROP INDEX city_name_index
ON city;
```

Indekser – forts.

- Plutselig går spørringen mer enn 100 ganger så fort!
  - Hva skjer!?

- Når vi lager en indeks, lagres en ekstra blokk data som optimaliserer WHERE clause delen av en spørring mot en (eller flere) kolonner.
  - Trenger én indeks per kolonne(-kombinasjon) vi ønsker å optimalisere en WHERE clause imot.
  - Merk: Indekser oppdateres når rader legges til eller fjernes. Endringer av innhold tar derfor litt lengre tid om vi har indekser på tabellen.
  - Vanligvis gjør vi mange flere spørringer enn endringer, så indekser er som regel verdt det allikevel. (Og oppdateringene tar ikke 100 ganger lengre tid.)

Transaksjoner

- Transaksjon: (i databasesammenheng)
  - "En handling eller serie handlinger, utført i sammenheng av en bruker eller et program, som leser eller endrer innholdet i en database."

- Eksempel på handling/serie handlinger:
  - CRUD: Create, Read ("select"), Update, Delete

Start transaction, commit og rollback

- DBMSen ("databasen") klarer ikke å ha noen formening om hvilke CRUD handlinger som hører sammen i én transaksjon.

- Vi forteller derfor DBMSen når en transaksjon starter ved å bruke START TRANSACTION.
  - Om alle operasjoner i transaksjonen lykkes, avslutter vi med COMMIT for å permanent lagre alle endringene.
  - Om noen av operasjonene feiler underveis, avslutter vi med ROLLBACK for å tilbakestille alle endringene.

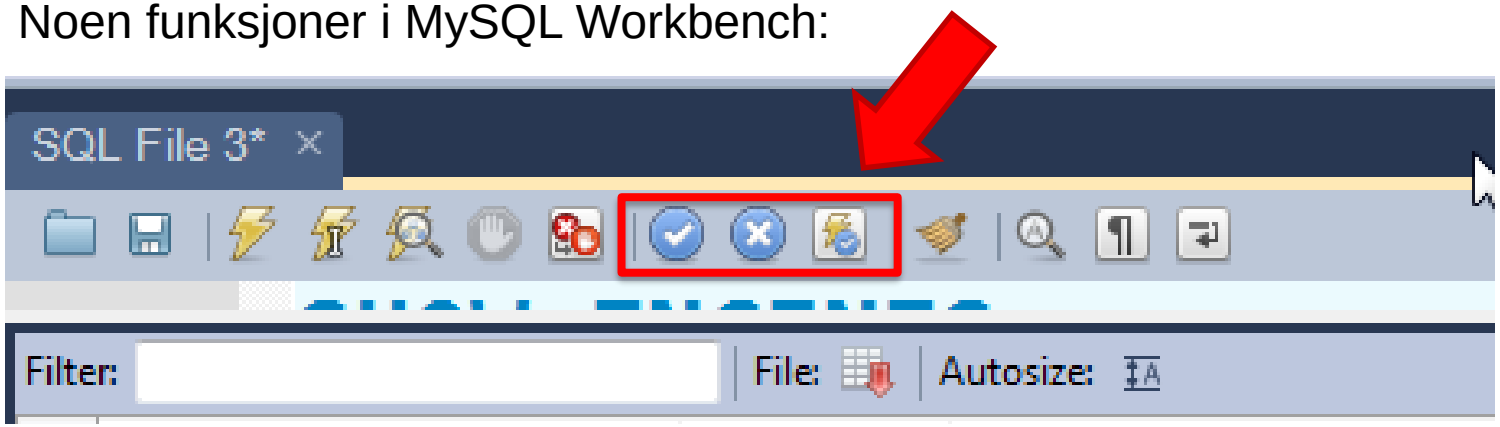
Commit og rollback – forts.

- For transaksjoner som bare krever én operasjon er dette unødvendig tungvint, og vi kan bruke "autocommit".

- MySQL Workbench har autocommit som default, men dette kan slåes av og på i toolbar'en. ("Toggle autocommit mode".)

Autocommit i MySQL Workbench

- Noen funksjoner i MySQL Workbench:



Toggle autocommit: If selected, each statement will be committed independently.

Autocommit i MySQL Workbench – forts.

Commit: Commits the current transaction.

Rollback: Rolls back the current transaction.

Autocommit i MySQL Workbench – forts.

- **Note:** All query tabs in the same connection share the same transactions.
  - To have independent transactions, a new connection must be opened.

- For eksempel på bruk av transactions i MySQL Workbench, se video i spillelisten, der jeg viser dette.
  - Da får du i samme slengen også sett et eksempel på forskjellig DB-engines, og hvordan vi kan endre disse!

ACID-egenskapene

- ACID-egenskapene er et knippe egenskaper alle transaksjoner bør oppfylle.

- Egenskapene har fått navnet **ACID** etter første bokstav i hver av de:
  - **A**tomicity
  - **C**onsistency
  - **I**solation
  - **D**urability

ACID-egenskapene – forts.

- Atomicity:
  - "alt eller ingenting" prinsippet: Enten gjennomføres en hel transaksjon, eller så tilbakestilles alt.
- Consistency:
  - En transaksjon må flytte databasen fra én fullverdig tilstand til en annen.
  - Dette ansvaret hviler på både utvikler og DBMS.
- Isolation:
  - Det som skjer internt i en transaksjon skal være usynlig for omverdenen (usynlig for andre transaksjoner) inntil transaksjonen er fullført.
- Durability:
  - Resultatet av en fullført transaksjon skal lagres i databasen, uavhengig av hva som skjer i kommende transaksjoner.

DBMS brukeradministrasjon

Fra emnebeskrivelsen:

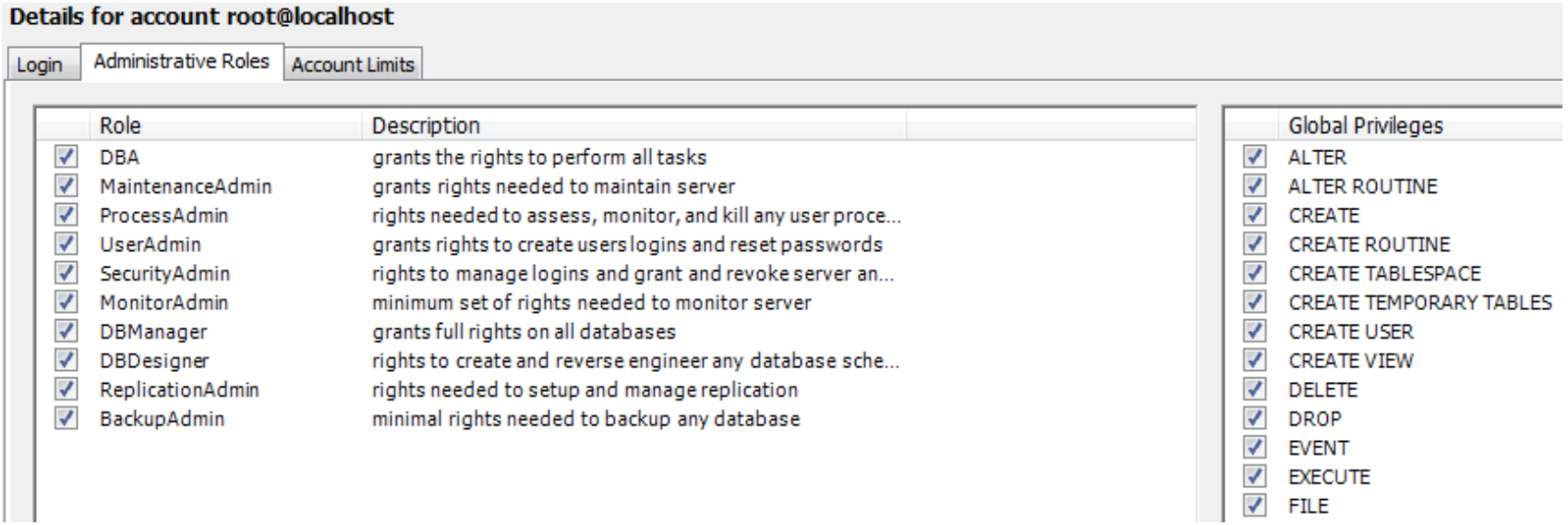
- "Etter å ha fullført emnet skal studenten kunne:
  - forklare roller og rettigheter.
  - utføre enkel brukeradministrasjon."

DBMS brukeradministrasjon – forts.

- En bruker kan gis en eller flere roller, og ha rettigheter knyttet til roller.
- Formålet er at en person skal kunne utføre de oppgavene han/hun er ment å utføre, men ikke mer enn det.
- Vi kan også gi rettigheter til ulike funksjoner i SQL, og til spesifikke databaser/tabeller.

DBMS brukeradministrasjon – forts.

- Root brukeren kan gjøre alt, og kan derfor gi alle mulige rettigheter til andre:



DBMS brukeradministrasjon, eksempel

- *Eksempel:* Vi skal lage en bruker som skal kunne utføre alle typer SQL-statements, *unntatt drop,* mot databasen world.

- Vi må i denne sammenheng:
  - Opprette bruker.
  - Gi (begrense) brukeren riktige rettigheter.
  - Opprette en ny forbindelse (kopling) til databasen der vi logger inn som den nye brukeren.
  - Sjekke at rettighetene stemmer.

- Dette kan du se i denne videoen (som ligger i spillelisten for emnet).

Om å gi rettigheter

- I eksempelet benyttet jeg grensesnittet til MySQL Workbench for å legge til rettighetene. Vi kan også gjøre dette via SQL.

- *Eksempel:* Gi SELECT rettighet til brukeren student (på localhost), for tabellen country i world databasen:

```
GRANT SELECT
ON world.country
TO 'student'@'localhost';
```

Backup

- Gjerne foretatt av en db-administrator.
- Backup-rutiner går kontinuerlig. Disse rutinene er for noen bedrifter helt kritiske for bedriftens eksistens.
- Fysisk backup skal ikke ligge i samme bygg som db!
  - Kan per i dag gjerne lagres i skyen.

Personopplysningsloven

- Personopplysningsloven er pensum, men står ikke i læreboka.
  - (For kilder, se senere slide.)

- Fra emnebeskrivelsen:
  - "Etter å ha fullført emnet skal studenten kunne forklare hvilke lover og regler som gjelder for lagring og bruk av personopplysninger i Norge."

Personopplysningsloven

- Personvernspørsmålene ifbm. bruk av IT dukket opp på 70-tallet.
  - Personregisterloven ble vedtatt i 1978.
  - Bl.a. etablerte loven Datatilsynet.
  - I 2001 kom personopplysningsloven, som en erstatter for den gamle.

Personopplysningsloven – forts.

- Et hovedprinsipp i personopplysningsloven er at du i større grad skal ha kontroll med opplysninger om deg selv.
  - Du har bl.a. krav på å få vite hvilke opplysninger en hvilken som helst virksomhet har om deg, hvor den har de fra og formålet med disse, innen 30 dager fra en henvendelse.

- Loven gjelder både elektroniske og manuelle registere, så lenge de er knyttet til en person.

Personopplysningsforskriften

- Personopplysningsforskriften utfyller personopplysningsloven.

- Den utdyper nærmere temaer som:
  - Konfidensialitet: Beskyttelse mot at uvedkommende får innsyn.
  - Integritet: Beskyttelse mot uautorisert endring.
  - Tilgjengelighet: Tilstrekkelig og relevant info.

Personopplysningsforsk. – forts.

- Noen av temaene fra personopplysningsforskriften :
  - Sikkerhetstiltak skal være planlagte og systematiske. (Tiltakene skal dokumenteres.)
  - Den som har den daglige ledelsen av virksomheten har ansvaret.
  - Det stilles krav til risikovurdering. (Sannsynlighet for og konsekvenser av sikkerhetsbrudd.)

Kilder: *regjeringen.no* –> *personregisterloven*
personopplysningsloven
personopplysningsforskriften

GDPR

- GDPR står for: **G**eneral **D**ata **P**rotection **R**egulation.

- Gjelder alle som har med EU land å gjøre, og Norge gjennom EØS.
  - Ble innført sommeren 2018.
  - Stor påvirkning på hvordan norske (og utenlandske) bedrifter håndterer data.

- Videoforklaring på 3 minutter (ligger i spillelisten).

- Det kan svi å ikke følge GDPR (29 mrd kroner hittil)

Neste gang

- Forberedelse til eksamen!