# Eksamen med LØSNING i 2DAN201-1 19H Databaser og nettverk

Dato: 4.12.2019

Tid: 09:00 til 13:00

Tillatte hjelpemidler:

* Lærebok og andre skriftlige dokumenter
* Egen datamaskin.

Det forutsettes at du har følgende programmer (eller tilsvarende) installert på PCen din: WAMP (Apache, MySQL eller MariaDB), MySQL Workbench, samt en php-eidtor (f.eks. Notepad++ eller VSCode, men Notepad kan også brukes). De som mangler dataprogrammer for å få utført oppgaven, kan levere hele eller deler av besvarelsen på papir.

Det er ikke lov å bruke internett, unntatt til å gå inn på Inspera. Praktisk utførelse av dette:

* Du kobler deg opp mot Inspera for å hente ut oppgavetekst og noen oppgitte data.
* Så slår du av internett på PCen (SLÅ PÅ FLYMODUS)
* Når oppgaven skal leveres, slår du på internett for å bruke Inspera

Et par ting dere må passe på:

* Når dere lager en tabell, sett alltid Engine til InnoDB (de fleste har det som default). Hvis det er problemer, sjekk at dere har engine InnoDB på alle tabeller som er involvert.
* Når Workbench lager en index, er det i scriptet med ordet VISIBLE etter hver index. MariaDB og MySQL liker ikke den, så den må slettes (kommer an på versjon av databasene)
* Dersom dere lager en FK med flere felter, må den tabellen som refereres ha en index på disse feltene med samme rekkefølge på feltene som det er i din FK.

OPPGAVE 1 35%

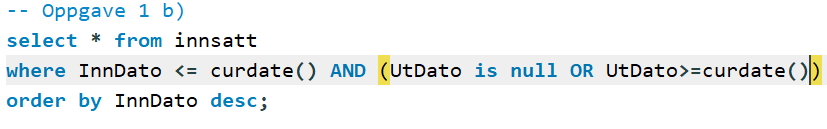
Her er en beskrivelse av et databasesystem for å holde orden på innsatte i fengsler. Det følger med ei script fil som lager denne databasen. NB! Sone i denne sammenhengen betyr å sitte i fengsel (sone straffen sin), og har ingenting med at fengselet er delt opp i soner.

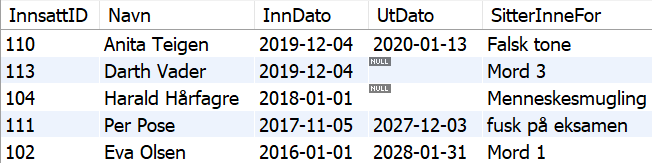
Databasen inneholder følgende tabeller:

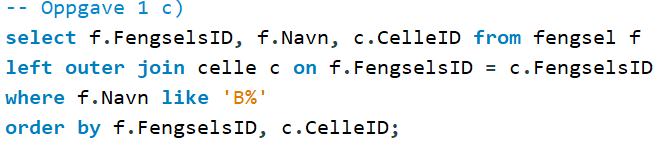
* **Fengsel** – FengselsID og navn på fengslene
* **Celle** – CelleID samt hvilket fengsel cella er i (celler er nummerert lokalt for hvert fengsel)
* **Innsatt** – ID og navn på innsatt, samt dato for soning. InnsattID er nummerert slik at de er unike uavhengig av fengsel (dvs. ikke lokale for hvert fengsel)
* **SonerICelle** – kobling mellom innsatt og celle, slik at en ser i hvilken celle og fengsel en innsatt sitter i.

Opprett databasen prison, med tabeller og data fra medfølgende script

Lag SQL-setninger som gjør følgende:

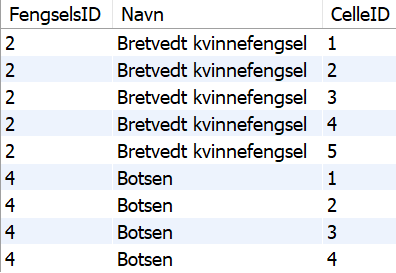
1. Lister opp alle innsatte. 1a) select \* from innsatt;
2. Lister opp alle innsatte (alle i tabellen innsatt) som sitter inn nå, sortert synkende på datoen de ble satt inn (innsatt.InnDato)



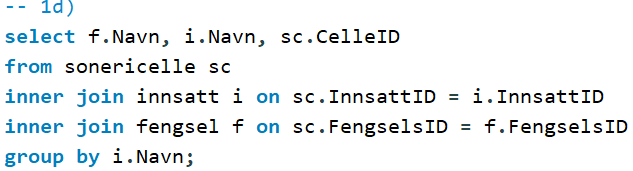


1. Lister opp alle fengsler som begynner på boksatven B, med tilhørende celler, sortert på fengselsid og celleid

LØSN: Det er greit med INNER JOIN, men da får en bare med de fengslene som har registrerte celler (men hva skal en med et fengsel uten celler?)



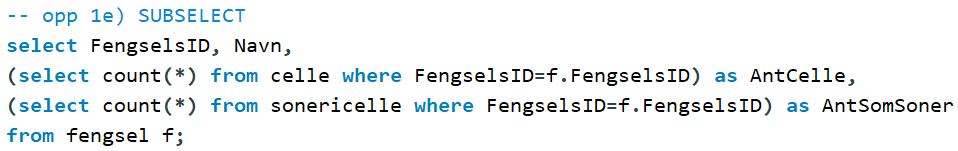
1. Lister opp alle innsatte, med navn på fengsel og person, og id på celle

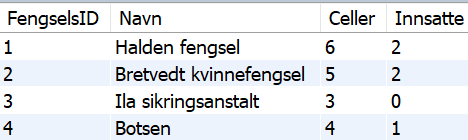




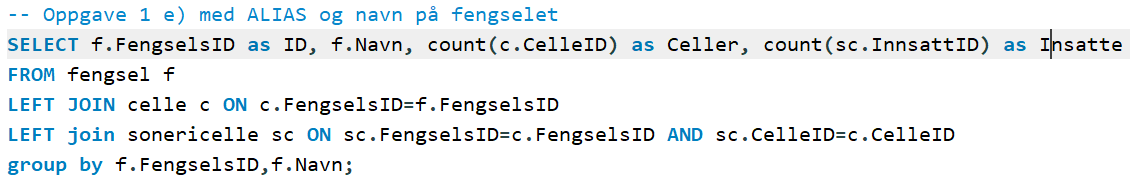
1. Lister opp alle fengsler (en rad pr. fengsel), med antall celler pr fengsel og antall innsatte pr fengsel.

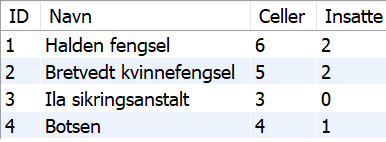
LØSNINGSFORSLAG:





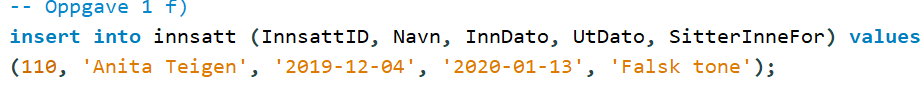
Denne SELECT vil også fungere, med outer join



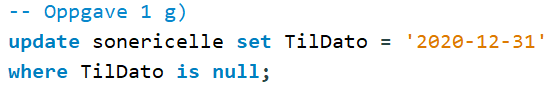


1. Setter inn en rad i tabellen innsatt

Løsningsforslag: Dersom en ikke har InsattID som autoincrement, men en angi en unik verdi for InsattID.



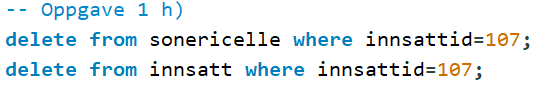
1. Oppdaterer tabellen sonericelle med TilDato '2020-12-31' på alle rader som mangler TilDato



1. Sletter en innsatt som er ferdig med soningen

LØSN: Fordi sonericelle har en FK som kobler InnsattID mot tabellen Innsatt, må en først slette raden i conericell før en kal slette den innsatte i tabellen innsatt.

Når en innsatt slipper ut (eller flyttes), er det naturlig å sette en tildato istedenfor å slette raden. Men etter ei stund kan det hende dataene skal slettes. Og dersom det er lagt inn noe feil, bør en kunne slette.



OPPGAVE 2 35%

* Oppdater databasen fra oppgave 1 slik at databasen kan brukes til å holde rede på ansatte også. Fengslene er delt opp i avdelinger. Cellene skal kobles til avdelinger
* Lag en vaktliste tabell, som en slags logg over hvem (ID for ansatt) som har vært på vakt på hvilken avdeling, og når han/hun hadde vakt (fra og til dato og klokkeslett). Denne tabellen kan også fylles ut på forhånd, og dermed brukes til å se hvem som skal på vakt.

Oppdater databasen fra oppgave 1 slik at databasen kan brukes til å holde rede på ansatte også. Fengslene er delt opp i avdelinger. Basen skal kunne inneholde ei vaktliste om når den enkelte skal jobbe, og på hvilken avdeling han/hun skal jobbe. Cellene skal også kobles til avdelinger.

1. Lag ei spørring som viser hvem som jobber på hvilke avdelinger for et angitt tidspunkt.
2. Lag ei spørring som returnerer navn på både innsatte og ansatte, med en kode e.l. som forteller om det er en ansatt eller en innsatt.

LØSNINGSFORSLAG: 3 tabeller:

* Ansatt – med data om de ansatte. Et alternativ er å utvide tabellen Innsatt til å inneholde ansatte også, og kalle den person e.l. Men det er gode grunner til å holde innsatte og ansatte adskilt, også i basen.
* Avdeling – informasjon om hver enkelt avdeling. Det er 2 måter å navngi/nummerere avdelingene på, enten pr fengsel eller fortløpende uavhengig av fengsel.
* Vaktliste – holder rede på hvem som har vakt (er på jobb) på hvilken avdeling for hvilke tidsrom.
* I tillegg må tabellen celle utvides med AvdelingsID.

Punkt en er å plassere data i riktig tabell. Det naturlige er at ansatte er ansatt i et bestemt fengsel. For store fengsler med store avdelinger kan en kanskje tenke seg at en ansatt er knyttet til en bestemt avdeling. Men de er ikke ansatt i en bestemt celle, så CelleID i ansatt tabellen blir veldig galt.

Typer er også viktig. IDer skal være heltall. Dato og klokkeslett skal være Date, DateTime eller Time.

Alle tabeller skal ha en fornuftig PK

Faste koblinger i basen bør ha FK, for å sikre datakvalitet:

* I Avdeling bør FengselsID vær FK mot tabellen Fengsel
* I Celle bør AvdelingsId vær FK mot tabellen Avdeling
* I Vaktliste bør FengselsID, AvdelingsID vær FK mot Avdeling og AnsattID vær FK mot tabellen Ansatt.

OPPGAVE 3 30%

1. Forklar i grove trekk hva en brannmur (firewall) gjør.

LØSN: En brannmur regulerer hva som kan koble seg ut eller inn på PC’en gjennom nettverket. F.eks kan man sette visse programmer til å ikke kunne kommunisere utad.

1. Kan du nevne et par fordeler et firma kan ha ved å legge applikasjoner på en skyløsning, framfor å kjøre dem på egne servere?

LØSN: Skalering, slipper stor investering i forkant (sammenlignet med å ha egne servere), slipper drifting av grunnsoftware (kommer an på skyløsning, om det er SaaS etc.), tilgang over alt, dataene er der om harddisken din krasher (en del av drifsfordelene: lagres flere steder, backup blir gjort)

LØSN: Det koster mye penger og kunnskap å sette opp maskinvare og serverprogramvare, og sørge for at det skalerer riktig når trafikken øker. Derfor er det mye mer praktisk å leie kapasitet fra de store leverandørene hvor man kun betaler for faktisk bruk, og man trenger ikke sette opp noe maskinvare og installere programvare selv. Tilbyderen har også da mer ansvar for sikkerheten

1. Hva er vranglås (deadlock) i databasesammenheng?

LØSN: Bør nevne transaksjon (ellers - 1)

En prosess(trå, connection) som har låst a og venter på b, mens en annen connection har lårt b, men venter på å få låse a.

1. Hvorfor er implementasjonen av nettverk (f.eks. på PCen din) organisert i lag?

LØSN: Hvert lag har sitt ansvarsområde, og sine protokoller. Det øker fleksibiliteten både ved bruk, og kanskje særlig ved endring. Endringer i en protokoll kan gjøres i ett lag, uten å røre de andre. Innføring av nye protokoller, som f.eks. IPv6 berører bare ett lag (i hovedsak, av og til må en også gjøre justeringer på lagene rundt). Dersom en lager et program som skal kommunisere over nettet på en annen måte enn det applikasjonslaget i TCP/IP stacken tilbyr av tjenester, er det også mulig å programmere direkte mot en av lagene lenger ned i stacken. Det er generelt ofte lurt å ha en viss lagdeling i programmer.

1. På de fleste hjemmenettverk får PCer etc. automatisk en Ipv4 adresse. Når du på nettet sjekker hvilken Ipv4-adresse du har, er det oftest en annen en det PCen din har intern i nettverket. Hva heter den mekanismen som gjør at intern og ekstern Ipv4 adresse er forkjellig, og hva er hensikten med denne mekanismen?

LØSN: Der er flere ting involvert, men hovedmekanismen som gjør dette er NAT-ing (Network Address Transfer), en funksjonalitet som ofte finnes i routeren.

1. Forklar oppdelinga av IP-adresser i en nettverksdel og en host-del

LØSN: Nettverksdelen sier hvilket nettverk PCen tilhører, det er den som routes på nettet.

Host-delen brukes til å nummerere Pcer innenfor et nettverk, slik at IP-adressen blir unik.

1. Hva er den største fordelen ved å gå over fra Ipv4 til Ipv6? LØSN: Antall adresser
2. En PC har IPv4 adressen 172.16.254.1/24. Hvilke av Pc-ene i) til v) er i samme nettverk som denne? i)172.16.254.5 ii)172.16.1.1 iii)172.16.254.4 iv)172.6.254.128 v) 72.16.254.1

LØSN: i og iii

1. Forkort denne IPv6 adressen så mye som mulig: 2001:0000:0000:aaaa:0001:0000::

LØSN: 2001:0:0:aaaa:1::

..aaaa:1:0:: Trekk 1. Ikke forkortet 0001 = trekk 1 Ikke forkortet 0000:0000 Trekk 1

1. En del Ipv6 adresseområder er reservert for spesiell bruk. En av dem er link-local, som brukes internt i nettverket. Om link-local står det:

«In IPv6, the block fe80::/10 is reserved for IP address autoconfiguration.[6] The implementation of these link-local addresses is mandatory, as various functions of the IPv6 protocol depend on them.»

Spørsmål: Er adressen feb9:3456:0:af03::ffe5:2f4 ei link-local adresse etter denne definisjonen? Begrunn svaret.

LØSN: Svaret er JA

fe80::/10 – de 10 første bitene er: 1111 1110 10 - slik må det se ut for at det skal være link local

De 10 første bitene av feb9:3456:0:af03::ffe5:2f4 – 1111 1110 10 - som stemmer med kravet