IN2090 – Innlevering 1
SQL 1

Publisert dato: 10.09.2026.

Innleveringsfrist: 24.09.2026 23:59.

Filer som skal leveres: 1: `oblig1.sql`.

Antall forsøk: 2.

Les gjennom hele teksten nøye før du begynner å løse oppgavene. Oppgaven skal leveres som en ren tekstfil med filendelsen `.sql`. Bruk SQL-kommentarer for ren tekst og kommentarer. Merk at oppgave 1 MÅ være gjort og du må ha levert en fil i Devilry for å få godkjent denne obligen og kunne gå opp til eksamen. De andre oppgavene er sterkt anbefalt å gjøre, men du trenger derimot ikke riktig svar på disse. Bruk disse til å teste egen kunnskap, og gjerne still spørsmål til retterne om det er noe du synes er vanskelig eller trenger hjelp til.

Bakgrunn og data for oppgavene

Et romskip ble observert flyvende forbi Jorden rundt semesterstart, og Departementet for Hemmelige, Utenomjordiske og/eller Farefulle Fenomener (HUFF) ønsker nå å finne ut hvor disse romvesnene kan ha kommet fra, altså hvilket solsystem og hvilken planet. De ønsker også å undersøke eventuelle muligheter for liv på andre planeter. HUFF er en svært hemmelig organisasjon, som sitter på svært mye informasjon, også dine formidable forelesningsnotater, svar på ukesoppgaver, tanker og idéer fra ditt arbeid med IN2090 dette semesteret. Imponert over dine evner i SQL, ønsker de derfor hjelp fra akkurat deg til dette!

De trenger hjelp med å hente ut informasjon fra et stort datasett de har fått samlet sammen. Databasen de har beskriver solsystemer, deres planeter, samt hvilke stoffer/molekyler som er oppdaget på disse planetene[1].

Databasen består av tre tabeller:

```
Stjerne(navn, [avstand], [masse])

Planet(navn, [masse], [oppdaget], [stjerne])

Materie(planet, molekyl)
```

Understrekede attributter angir primærnøklene. Attributter i klammeparenteser er attributter som ikke nødvendigvis har data, dvs. de kan inneholde `NULL`.

Tabellen `stjerne` beskriver stjerner, som identifiseres med deres `navn`, og har kolonnen `avstand` som inneholder stjernens avstand fra solen i parsecs, samt kolonnen `masse` som oppgir stjernens masse relativt til solen (hvor 1 betyr lik som solen, 2 betyr dobbelt så massiv som solen, 0.1 betyr en tiendedel av solens masse, osv.).

Tabellen `planet` beskriver planeter, som også identifiseres med dens `navn`, og inneholder også planetens `masse` relativt til Jupiter (den største planeten i vårt eget solsystem), et årstall som representerer når planeten ble `oppdaget`, samt en referanse til `stjernen` den går i bane rundt (altså er `planet(stjerne)` en fremmednøkkel som refererer til `stjerne(navn)`).

Tabellen `materie` beskriver hvilke stoffer som er oppdaget på hvilke planeter. Her er `planet` en fremmednøkkel som peker på `planet(navn)` og `molekyl` inneholder et molekyl som er oppdaget på planeten i `planet`-kolonnen. Merk at én planet kan ha mange molekyler, og kan derfor forekomme flere ganger i tabellen.

Oppgave 1

Oppgave 1 går ut på å kjøre et SQL-script som oppretter databasen du skal bruke i resten av oppgavene. For de resterende oppgavene skal du levere SQL-spørringene/kommandoene som løser oppgavene.

Opprett en database på ditt eget database-område ved å kjøre kommandoene under. I kommandoene må du bytte ut `brukernavn` med ditt UiO-brukernavn. I enkelte av kommandoene vil du bli bedt om å oppgi ditt UiO-passord.

For å sette opp databasen gjør du følgende:

[1] Dataene er ekte data om virkelige planeter og er hentet fra: `http://exoplanet.eu/catalog/`

(a) last ned filen `planeter.sql` fra

`https://www.uio.no/studier/emner/matnat/ifi/IN2090/h26/innleveringer/planeter.sql`

til en mappe.

(b) kjør scriptet i egen database, enten ved å logge inn (fra samme mappe som filen over) på PostgreSQL med

```
psql -h pg-in2090.dbd.uiocloud.no -U brukernavn -d brukernavn
```

(hvor du bytter ut `brukernavn` med ditt UiO-brukernavn) og så kjøre kommandoen `\i planeter.sql`; eller ved bruke `-f`-flagget med `psql` direkte, slik:

```
psql -h pg-in2090.dbd.uiocloud.no -U brukernavn -d brukernavn -f planeter.sql
```

Du har nå databasen satt opp med databasen beskrevet over og du kan nå kjøre SQL-spørringer over dataene.

Merk: Denne oppgaven MÅ være løst for at du skal få godkjent på denne obligen. Det blir undersøkt at din personlige database inneholder dataene fra scriptet når din besvarelse rettes. Dette verifiseres ved at følgende spørring kjøres, og gir ut 403 som resultat:

```
SELECT count(*) FROM materie;
```

For å få godkjent obligen må altså databasen være opprettet og du må ha levert en fil i Devilry hvor du sier at du har gjort oppgave 1.

Om du har problemer med `psql` eller å få kjørt scriptet over, kontakt din gruppelærer eller sjekk PostgreSQL-guiden[2].

Oppgave 2

Før HUFF er komfortable med å bruke dataene i databasen ønsker de å kontrollere korektheten av dataene ved å hente ut litt informasjon om noen solsystemer de allerede kjenner godt fra andre kilder. De ønsker også å vite hvor innholdsrik databasen er.

De ber deg derfor skrive SQL-spørringer som finner ut:

[2] `https://www.uio.no/studier/emner/matnat/ifi/IN2090/h26/programvare/postgres.pdf`

a) Navn på alle planeter som går i bane rundt stjernen med navn `Proxima Centauri`.

b) Hvilke årstall (uten duplikater) det ble oppdaget planeter i bane rundt stjernene med navn `TRAPPIST-1` og `Kepler-154`.

c) Antall planeter det er som ikke har oppgitt en masse (altså hvor masse er `NULL`).

d) Navn og masse på alle planeter oppdaget i 2020 med masse høyere enn gjennomsnittet av massen til alle planeter.

e) Antall år mellom første og siste oppdagede planet i `planet`-tabellen.

Oppgave 3

Basert på resultatet av spørringene dine er HUFF fornøyde med dataene og ønsker å gå videre med dem. De ønsker nå å begynne søket etter planeten som romvesnene som besøkte Jorden kom fra. De har ulike modeller som baserer seg på ulik informasjon, og din jobb blir å finne lister med planeter som passer hver modell. (Merk: Resultatene fra spørringene du skriver trenger ikke ha overlappende svar. Dersom det ikke er noen planet som passer alle modellene, kan det tyde på at romvesnene reiste fra en annen planet enn den de evolusjonært er tilpasset, altså at Jorden ikke er den første planeten de besøker.)

a) HUFF klarte å hacke datasystemene til romskipet, og kunne da hente ut en bildeserie med selfies fra romvesnene. Den ene modellen baserer seg derfor på anatomien til romvesnene: Biologer fra HUFF har nemlig funnet ut, basert på tykkelsen på bena og størrelsen på musklene, at de må komme fra en planet med masse mellom 3 og 10. Videre tror de at romvesnene inneholder vann (altså H2O), så planeten må inneholde dette molekylet. Skriv derfor en spørring som finner navn på alle planeter som passer denne informasjonen.

b) Den andre modellen baserer seg på romskipet og dens motor[3]: HUFFs ingeniører har nemlig funnet ut at romskipet høyst kan ha reist en avstand lik stjernens masse ganger `12`, altså må planeten gå i bane rundt en stjerne som har en avstand mindre enn `12` ganger dens masse. Videre har ingeniørene funnet ut at motoren bruker hydrogen som drivstoff, så planeten de reiser fra må ha molekyler som inneholder hydrogen

[3]For romreise bruker man ofte et stort objekts gravitasjon for akselerasjon.

(altså molekyler som inneholder bokstaven `'H'`)[4]. Skriv derfor en SQL-spørring som finner navn på planeter som passer informasjonen over.

c) Den tredje modellen er også laget av HUFFs ingeniører: Et alternativ til å bruke en massiv stjerne er å bruke to massive planeter for akselerasjon. Romvesnene kan ha reist fra en av to planeter fra samme solsystem, begge med masse større enn `10`, gitt at avstanden til deres solsystem er mindre enn `50`. Skriv en SQL-spørring som finner navn på alle planeter som passer med informasjonen over.

Oppgave 4

Nils, en av de ansatte på HUFF, ble svært imponert over hva du får til med SQL, og forsøkte seg derfor på å skrive en spørring selv. Nils ønsket å finne ut når planeter som går i bane rundt en stjerne med avstand større enn `8000` ble oppdaget, og skrev derfor denne spørringen:

```
SELECT oppdaget
FROM planet NATURAL JOIN stjerne
WHERE avstand > 8000;
```

Forklar Nils hvorfor spørringen hans ikke finner det han er ute etter og hvorfor den ikke gir noen rader i resultatet.

Oppgave 5

Datasettet brukt så langt inneholder kun stjerner og planeter utenfor vårt solsystem. HUFF ønsker nå at vår egen sol og egen planet (altså Jorda) skal legges inn i databasen, slik at disse kan bli med i fremtidige analyser og sammenlikninger. Skriv derfor SQL-kommandoer som:

a) Setter inn *Sola* i `stjerne`-tabellen, med `navn` lik `Sola`, `avstand` lik `0` og `masse` lik `1`.

b) Setter inn *Jorda* i `planet`-tabellen, med navn lik `Jorda`, `masse` lik `0.003146`, `oppdaget` lik `NULL`, og `stjerne` lik `Sola`.

Oppgave 6

Nå som liv utenfor Jorden er bekreftet av de besøkende romvesnene, ønsker HUFF å utvide søket etter andre livsformer i rommet. De ønsker nå å lagre

[4]Merk: Dette vil også inkludere `'He'`, altså helium, som jo ikke er hydrogen, men du trenger ikke ta hensyn til dette.

observasjoner av mulige tegn til liv i en egen tabell med navn `observasjon`, og ønsker hjelp av deg til å lage denne tabellen. Hver observasjon skal identifiseres med en egen `observasjons_id` som er et heltall. Hver observasjon skal også ha et tidspunkt med dato (altså et `timestamp`) knyttet til seg, samt en referanse til planeten observasjonen ble gjort på. Videre ønsker de å kunne lagre en kommentar til observasjonen som da bare er ren tekst (men en observasjon trenger ikke ha noen slik kommentar). Skriv en SQL-kommando som lager denne tabellen.

Levering

Den obligatoriske oppgaven leveres i Devilry[5]. Sørg for at du er registrert i systemet ved å logge inn og se at `oblig1` er tilgjengelig som en oppgave for IN2090. *Sjekk dette før du begynner å løse oppgavene!*. Dersom du ikke er registrert, send en mail til `leifhka@ifi.uio.no`.

Du kan levere så mange ganger du vil, det er kun den siste leveringen som teller. Alle leveringer som blir lastet opp etter fristen vil ikke bli godkjent med mindre man har en godkjent grunn for å levere sent, se mer informasjon om dette på IFIs nettsider[6].

Lever kun en rentekstfil `oblig1.sql`. En rentekstfil er altså det vi lager i en kodeeditor, f.eks. Emacs, Vim, Notepad, Atom, VSCode, osv.

Lykke til!

[5] `https://devilry.ifi.uio.no/`

[6] `https://www.uio.no/studier/admin/obligatoriske-aktiviteter/mn-ifi-oblig.html`