IN2090 – Innleveringsoppgave 5
Programmering med SQL

Publisert dato: 05.11.2026.

Innleveringsfrist: 19.11.2026 23:59.

Filer som skal leveres: 1: `huffsa.py` eller `Huffsa.java`.

Antall forsøk: 1.

Les gjennom hele teksten nøye før du begynner å løse oppgavene. Du kan velge om du ønsker å løse oppgaven med Python eller Java.

Etter din hjelp til analyser av folks følelser, har HUFF nå konkludert med at menneskeheten er klar for økt kontakt med, og kunnskap om, romvesener. Romvesnene som landet på Blindern for noen uker siden (de vi kommuniserte med i oblig 1) har dessverre for lengst reist sin vei, og HUFF er derfor avhengig av å finne nye romvesner de kan kontakte og formidle kunnskap om.

De har derfor startet et stort prosjekt, HUFFs Søk etter Aliens (HUFFSA) for å søke etter liv på andre planeter. Prosjektet kommer til å involvere mange mennesker (og andre enkelt individer), og de fleste av disse kan dessverre ikke (ennå) SQL. Prosjektet er todelt, hvor den første delen går ut på å finne planeter hvor det kan forekomme liv, og del to går ut på å forsøke å kommunisere med eventuelle vesener på de gitte planetene og dokumentere resultatene av disse forsøkene. Hver del trenger derfor skreddesydd programvare som de ansatte kan bruke for å søke i databasen etter planeter (oppgave 1), og for å sette inn resultatene av forsøkene (oppgave 2).

I oppgavene skal du skrive funksjoner/metoder som jobber mot din private database (altså den som har ditt UiO-brukernavn som navn), og du skal bruke `_priv`-brukeren som du har fått informasjon om i egen epost (brukernavnet er ditt UiO-brukernavn etterfulgt av `_priv`). Så f.eks. om du ønsker å logge inn i denne databasen med denne brukeren kan du kjøre:

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

hvor du erstatter `brukernavn` med ditt UiO-brukernavn. Merk at programmene du skriver (i enten Java eller Python) skal også bruke denne brukeren. Dette er fordi du skal slippe å lime inn ditt personlige UiO-passord i kildekoden.

Last ned Zip-filen[1] og unzip den (f.eks. med `unzip huffsa.zip`). Kjør SQL-scriptet `huffsa.sql` (fra den unzippede mappen) som setter opp databaseskjemaet som beskrevet i PostgreSQL-guiden på semestersiden.[2]

Dataene databasen din skal inneholde er lik de fra `planeter.sql` som du benyttet i Oblig2, men hvor det er lagt til noen ekstra kolonner i `planet`-tabellen for dokumentasjon av resultater på forsøkene beskrevet over. SQL-scriptet `huffsa.sql` som ligger i Zip-filen inneholder all data med disse endringene. De ekstra kolonnene som er lagt til er som følger:

- `liv`: sann hvis det er bekreftet liv på planeten, usann hvis ikke
- `skummel`: `NULL` hvis det ikke er oppdaget noe liv på planeten (ennå), sann hvis livsformen som er oppdaget ansees å være skummel, usann hvis ikke (eller `NULL` om det er ukjent om livsformen er skummel)
- `intelligent`: `NULL` hvis det ikke er oppdaget noe liv på planeten (ennå), sann hvis livsformen som er oppdaget ansees å være intelligent, usann hvis ikke (eller `NULL` om det er ukjent om livsformen er intelligent)
- `beskrivelse`: `NULL` hvis det ikke er oppdaget noe liv på planeten (ennå), ellers en tekstlig beskrivelse eller begrunnelse av farlig/intelligent-verdiene

Kjør dette scriptet i din personlige database (slik vi gjorde med `planeter.sql` i oblig 2), men kjør den med den overnevnte priv-brukeren. Da vil denne brukeren få alle tilgangene den trenger til de ulike tabellene.

I mappen du laster ned er det også et skjelett-program (`python/huffsa.py` for Python og `java/Huffsa.java` for Java) som du skal fylle ut i oppgavene under. I programmet må du sette `user` lik din vanlige UiO-bruker (`_priv` blir lagt til av programmet) og `pwd` lik det passordet du fant for `_priv`-brukeren i din personlige database.

Python

Dersom du velger Python må programmene du skriver kjøres med Python 3. Dette gjøres på følgende måte:

[1] `https://www.uio.no/studier/emner/matnat/ifi/IN2090/h26/innleveringer/huffsa.zip`

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

```
python3 huffsa.py
```

For Python kan det være (om du får en feilmelding om at `psycopg2` ikke finnes) du også må installere biblioteket `psycopg2`, som gjøres ved å kjøre:

```
pip3 install --user psycopg2-binary
```

Java

Dersom du velger å løse oppgaven med Java må du ha med filen `postgresql.jar` i *classpath*'en når du kjører programmet. Filen ligger i samme mappe som `Huffsa.java`. Du kjører så Java programmet (etter vanlig kompilering med `javac`) slik i Linux/Mac:

```
java -cp .:postgresql.jar Huffsa
```

og slik i Windows:

```
java -cp ".;postgresql.jar" Huffsa
```

Kjøring av programmer og fjerninnlogging

Programmene dere skriver skal kommunisere med databasen, og må derfor kjøres innenfor IFIs WiFi, på samme måte som ellers når dere logger inn i databasene. Altså må dere enten være fysisk på IFI, eller fjerninnlogge og kjøre programmene der. Dersom dere ønsker å både skrive og kjøre programmene via fjerninnlogging kan dere kjøre følgende kommando fra IFI-maskinen (etter fjerninnlogging) for å laste ned filene direkte til ditt UiO-hjemmeområde:

```
wget https://www.uio.no/studier/emner/matnat/ifi/IN2090/h26/innleveringer/huffsa.zip && \
unzip huffsa.zip && \
cd huffsa
```

Dersom du ønsker å skrive programmene lokalt på din maskin, og kun kjøre dem via fjerninnlogging, kan du flytte filene fra din maskin til UiO-hjemmeområdet ditt slik:

1. Først flytt alle filer du ønsker å flytte over i en egen mappe, f.eks. `move`
2. Zip-mappen med

   ```
   zip -r move.zip move
   ```

3. Flytt mappen til UiO med:

```
scp -J <brukernavn>@login.uio.no move.zip <brukernavn>@login.ifi.uio.no:~/
```

4. Logg inn på IFI med vanlig SSH-kommando:

   ```
   ssh -J <brukernavn>@login.uio.no <brukernavn>@login.ifi.uio.no
   ```

5. Pakk ut mappen med:

   ```
   unzip move.zip
   ```

6. Hopp inn i mappen med:

   ```
   cd move
   ```

Oppgave 1 – Finne planeter

I denne oppgaven skal du implementere funksjonen `planet_sok(conn)` (for Python) eller metoden `planetSok(Connection)` (for Java) som lar HUFF-ansatte søke etter planeter med mulig liv. Funksjonen skal la brukeren søke etter planeter basert på hvilke molekyler som er oppdaget på planeten, hvor de skal oppgi minst ett molekyl, men kan gi to molekyler. Dersom to molekyler er oppgitt skal programmet finne de planetene som har begge molekylene. Funksjonen skal til slutt printe ut alle planeter som tilfredsstiller søket, på formen:

```
Navn: <navn>
Planet-masse: <pmasse>
Stjerne-masse: <smasse>
Stjerne-distanse: <distanse>
Bekreftet liv: <liv>
```

hvor `<navn>` er planetens navn, `<pmasse>` er planetens masse, `<smasse>` og `<distanse>` er massen og distansen til planetens stjerne, og `<liv>` er `Ja` dersom det er bekreftet liv på planeten og `Nei` hvis ikke. Resultatet skal være sortert på stjernens avstand fra minst til størst. Du kan anta at brukerne alltid oppgir en verdi for `Molekyl 1`.

F.eks. kan output se slik ut (brukerens input er markert i rødt):

```
--[ HUFFSA ]--
Vennligst velg et alternativ:
```

```
 1. Søk etter planet
 2. Legg inn forsøksresultat
 3. Avslutt
Valg: 1
--[ PLANET-SØK ]--
Molekyl 1: C2H2
Molekyl 2: K

--Planet--
Navn: 55 Cnc e
Planet-masse: 0.02703
Stjerne-masse: 1.015
Stjerne-distanse: 12.34
Bekreftet liv: Nei

--Planet--
Navn: HD 209458 b
Planet-masse: 0.69
Stjerne-masse: 1.148
Stjerne-distanse: 47.0
Bekreftet liv: Nei

--Planet--
Navn: WASP-80 b
Planet-masse: 0.554
Stjerne-masse: 0.58
Stjerne-distanse: 60.0
Bekreftet liv: Nei
```

Dersom brukeren ikke oppgir noe i filteret på det andre molekylet (altså lar feltet stå blankt) skal programmet ikke gjøre noen begrensning på dette. F.eks. kan output da se slik ut (brukerens input er markert i rødt, merk ingen input for `Molekyl 2:`):

```
--[ HUFFSA ]--
Vennligst velg et alternativ:
 1. Søk etter planet
 2. Legg inn forsøksresultat
 3. Avslutt
Valg: 1
--[ PLANET-SØK ]--
Molekyl 1: O2
Molekyl 2:

--Planet--
```

```
Navn: GJ 1132 b
Planet-masse: 0.00522
Stjerne-masse: 0.181
Stjerne-distanse: 12.04
Bekreftet liv: Nei

--Planet--
Navn: HD 209458 b
Planet-masse: 0.69
Stjerne-masse: 1.148
Stjerne-distanse: 47.0
Bekreftet liv: Nei
```

Oppgave 2 – Sette inn forsøksresultater

I denne oppgaven skal du implementere funksjonen `sett_inn_resultat(conn)` (for Python) eller metoden `leggInnResultat(Connection)` (for Java) som lar brukeren sette inn nye forsøksresultater etter forsøk på kommunikasjon med en planet. Funksjonen skal altså oppdatere `planet`-tabellen basert på brukerens input. Du kan anta at brukeren alltid oppgir verdier for alle feltene, og at den alltid oppgir `j` eller `n` på `Skummel` og `Intelligent`.

En kjøring kan da se slik ut (brukerens input er markert i rødt):

```
--[ HUFFSA ]--
Vennligst velg et alternativ:
 1. Søk etter planet
 2. Legg inn forsøksresultat
 3. Avslutt
Valg: 2

--[ LEGG INN RESULTAT ]--
Planet: Kepler-265 d
Skummel: j
Intelligent: n
Beskrivelse: Svarte utelukkende med knurring og fresing.
Resultat lagt inn.
```

Etter dette skal raden med verdier

```
    navn     | masse | oppdaget |  stjerne   | liv | skummel | intelligent | beskrivelse
-------------+-------+----------+------------+-----+---------+-------------+-------------
 Kepler-265 d |       |     2014 | Kepler-265 | f   |         |             |
```

bli oppdatert til

```
    navn     | masse | oppdaget |  stjerne   | liv | skummel | intelligent |                beskrivelse
-------------+-------+----------+------------+-----+---------+-------------+-------------------------------------------
 Kepler-265 d |       |     2014 | Kepler-265 | t   | t       | f           | Svarte utelukkende med knurring og fresing.
```

Levering

Innleverinsoppgaver leveres i Devilry[3]. Sørg for at du er registrert i systemet ved å logge inn og se at `innlevering5` 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`.

Etter innlevering vil du få tilbakemelding på innleveringen din, i likhet med en oblig. Selvom innleveringsoppgaven ikke er obligatorisk, er den et svært nyttig læringstilbud og en viktig del av undervisningen i kurset. Fokuser på å lære mest mulig og les tilbakemeldingene du får fra retterne nøye. Merk også at det ikke blir publisert løsningsforslag på innleveringsoppgavene, så å levere dem inn og deretter få tilbakemelding er eneste måte å få kjennskap til hvor god egen besvarelse er.

Du kan levere så mange ganger du vil, men det er kun den siste leveringen du vil få tilbakemelding på.

Lykke til!

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