

# IN2090: Obligatorisk oppgave 6 (Python)

I denne oppgaven får du et delvis fullført program for å hente ut informasjon om timelister, og legge til nye timelistelinjer i en PostgreSQL-database. Oppgaven går ut på å fullføre dette programmet slik at programmet henter ut informasjon om timelister fra en database ved hjelp av pakken *Psycopg2*, og legger inn ny informasjon i databasen.

Koden du får utdelt inneholder en ferdigskrevet main-metode og kode for å koble deg til en database fra Python. Det du trenger å skrive, er metodene i *timelistedb.py*.

Start med å [laste ned nødvendige filer herfra](#). Pakk ut filene på et egnet sted. Du skal ha følgende filer:

- **timelistedb.py**: Det er i denne filen du skal skrive koden i denne obligen. Filen inneholder klassen *TimelisteDb* der metodene er deklarerert. Du skal skrive metodene så de gjør det de skal.
- **oblig6.py**: Dette er filen du skal kjøre, som kaller på metodene i *TimelisteDb*. Denne filen håndterer også tilkoblingen til PostgreSQL-databasen din på IFI fra Python. Her må du skrive inn ditt eget brukernavn, ellers trenger du ikke endre noe.
- **oppsett.sql**: Denne filen skal du importere i databasen din, på samme måte som du gjorde i oblig 3. Dette er en forenklet versjon av tabellene fra oblig 3, og inneholder tabellene `tliste` og `tlistelinje`.

Obligen skal skrives i Python 3 (ikke Python 2.7).

## Oppsett

1: Last ned og pakk ut filene i [oblig6-python.zip](#).

2: Du må være på UiO-nettet for å koble deg til PostgreSQL-databasen på IFI. Om du ikke befinner deg på UiO-nettet, må du sette opp en VPN-klient/program på PC-en din. [Instruksjoner for oppsett av VPN-klient/program finner du her](#) (NB: ikke bruk `view.uio.no`).

3: Bytt ut «dittBrukernavn» med brukernavnet ditt i *oblig6.py*.

4: Installer *psycopg2* ved å kjøre følgende kommando i terminalen:<sup>1</sup>

```
pip install psycopg2-binary
```

Deretter kan du prøve å kjøre programmet. Om du bruker en IDE som IDLE eller PyCharm, vil du skrive inn passordet i klartekst. For å omgå dette, kan du kjøre programmet i terminalen eller endre *oblig6.py* (se vedlegg for instruksjoner).

5: Importer *oppsett.sql* til databasen din, på samme måte som du gjorde i oblig 3: Kopier filen til et eget område på IFI, logg inn på PostgreSQL med `psql -h dbpg-ifi-kurs -U brukernavn`, og kjør kommandoen `\i oppsett.sql`. Alternativt kan du kopiere alle spørringene inn i `psql` for å opprette tabellene og sette inn data.

---

<sup>1</sup> Om du har både Python 2.7 og 3+ installert, kan det hende `pip` er for versjon 2.7 (se hvilken versjon av Python `pip` bruker ved å skrive `pip --version`). Da kan du bruke kommandoen `pip3` i stedet.

## Oppgave

Du skal fylle ut metodene i *timelistedb.py*.

**print\_timelister**(self)

Skal printe ut `timelistenr`, `status` og `beskrivelse` for alle timelistene i databasen. Du velger selv hvordan utskriften skal se ut, men ikke kjør `print()` direkte på tuppelet (du skal håndtere hvert attributt/kolonne for seg).

**print\_timelistelinjer**(self, `timelistenr`)

Skal printe ut `linjenr`, `timeantall`, `beskrivelse` og `kumulativt_timeantall` for alle linjer med `timelisteNr`. Du trenger ikke å skrive ut `timelistenummeret`. Du velger selv hvordan utskriften skal se ut, men ikke kjør `print()` direkte på tuppelet (du skal håndtere hvert attributt/kolonne for seg).

**median\_timeantall**(self, `timelistenr`)

PostgreSQL har ingen innebygget aggregeringsfunksjon for å regne ut medianverdi. Derfor ønsker vi å bruke Python til å hjelpe oss med utregningen. Metoden skal *returnere* (ikke skrive ut) median-timeantallet.

Medianen er tallet «midt på» listen av tall, dersom listen er sortert. Om det er et partall antall elementer, tar man gjennomsnittet av de to midterste tallene.

*Tips:* Du kan bruke `ORDER BY` i SQL-spørringen for å sortere etter timeantall. Mens du itererer over resultatene, kan du legge f.eks. legge verdiene i en ny liste. Bruk f.eks. [statistics.median](#) for dette (Python3.4+), ved å skrive `from statistics import median` øverst i fila, og deretter bruke `median(liste)`.

**sett\_inn\_timelistelinje**(self, `timelistenr`, `antall_timer`, `beskrivelse`)

Tar imot de tre parameterne `timelistenr`, `antall_timer` og `beskrivelse`, og kjører en `INSERT INTO`-spørring for å legge inn en timelistelinje med disse verdiene.

*Merk:* Du må først finne ut hva som er det høyeste linjenummeret, og i den nye linjen du legger inn må `linjeNr` være én høyere enn det høyeste. F.eks. om du har linjeNr 1-4, må den nye linjen ha linjeNr 5.

```
regn_ut_kumulativt_timeantall(self, timelistenr)
```

Et kumulativt timeantall er summen av timeantall *så langt i timelisten*. Du skal altså summere linjens timeantall med timeantall til alle linjer du allerede har iterert gjennom. For timeliste 2 vil kumulativt timeantall se slik ut:

linjenr	timeantall	kumulativt_timeantall
1	1	1
2	4	5
3	3	8
4	5	13

Dette er noe som kan være knotete å få til i SQL, men lett å gjøre i Python. Derfor skal du i denne oppgaven hente ut alle linjene med det oppgitte timelisteNr. Mens du itererer gjennom resultatene, skal du regne ut den kumulative verdien, og sette den tilbake inn i databasen ved hjelp av `UPDATE`-spørringer. Det er altså lurt å kjøre én `UPDATE` for hver rad i løkken.

## Innlevering

Siden du kun skal ha gjort endringer i *timelistedb.py*, er det kun denne filen du trenger å levere.

## Vedlegg: Fiks dersom `getpass` ikke fungerer

Om du kjører koden i en IDE som IDLE, PyCharm, Spyder e.l. som ikke skjuler passordet når du skriver det inn, kan du endre *oblig6.py* ved å bytte ut `getpass.getpass` med en `passwordbox` fra pakken *easygui*:

```
import pycpg2
import getpass
import sys
from easygui import passwordbox
from timelistedb import TimelisteDb

if __name__ == "__main__":
    if sys.version_info[0] < 3:
        print("OBS: Obligen skal løses med Python 3!")
        print("Kanskje det allerede er installert? Prov 'python3 oblig6.py'")
        sys.exit(1)

    conn = pycpg2.connect(
        host="dbpg-ifi-kurs.uio.no",
        dbname="dittBrukernavn", # sett inn ditt brukernavn her
        user="dittBrukernavn", # sett inn ditt brukernavn her
        password=passwordbox("Passord: "),
        sslmode='require')

    db = TimelisteDb(conn)
    ...
```

Du kan installere *easygui* med `pip`:

```
pip install easygui
```