

IN2090 – Ukesoppgaver 11

Programmering med SQL

Du velger selv om du vil løse oppgavene med Python (3) eller Java. Oppgavene denne uken uvider koden skrevet som del av denne ukens videoer, og koden er tilgjengelig fra timeplanen¹ i enkle filer, eller kan lastes ned som en Zip-fil² som inneholder SQL-scriptene, Python-koden, Java-koden og Javas avhengigheter.

Merk: Koden kobler seg til din personlige database på IFI, så i likhet med tilkobling til databasene via `psql` må man enten fjerninnlogge til en IFI-maskin, være på IFIs WIFI eller sitte på en IFI-maskin for å kunne kjøre koden.

Om du fjerninnlogger kan koden lastes ned i `bash` slik:

```
wget https://www.uio.no/studier/emner/matnat/ifi/IN2090/h20/ukesoppgaver/webshop.zip && \
unzip webshop.zip && \
cd webshop
```

For å få dataene inn i din personlige database, kjør de to SQL-scriptene `webshop.sql` og `data.sql` som ligger i Zip-filen. Disse kan kjøres ved å logge inn i din personlige database fra mappen med SQL-filene, og kjøre følgende fra `psql`:

```
\i webshop.sql
\i data.sql
```

Python

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

```
python3 user_frontend.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
```

¹<https://www.uio.no/studier/emner/matnat/ifi/IN2090/h20/timeplan/index.html#FOR>

²<https://www.uio.no/studier/emner/matnat/ifi/IN2090/h20/ukesoppgaver/webshop.zip>

Java

Dersom du velger å løse oppgaven med Java kan det være (om du får en `ClassDefNotFoundError`) at du må ha med filen `postgresql.jar` når du kjører programmet. Filen ligger i `webshop/java`-mappen i Zip-filen som man kan laste ned over. Du kjører så Java programmet (etter vanlig kompilering med `javac`) slik i Linux/Mac:

```
java -cp ".:postgresql.jar" UserFrontend
```

og slik i Windows:

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

Oppgave 1 – Forbedret søk

I denne oppgaven skal du forbedre søket etter produkter. Brukeren skal kunne sortere resultatene, og skal kunne velge om sorteringen skal skje på pris eller på produktnavn, i tillegg skal brukeren kunne velge om sorteringen skal skje fra minst til størst eller størst til minst. Til slutt skal brukeren kunne velge en begrensning på hvor mange produkter som skal vises. Merk at om brukeren ikke oppgir noe på spørsmålene, skal ingen sortering/begrensning skje. Et søk skal da kunne se slik ut:

```
-- USER FRONTEND --
Please choose an option:
 1. Register
 2. Login
 3. Exit
Option: 2
-- LOGIN --
Username: perh
Password: per123
Welcome Per Hansen
-- SEARCH --
Search:
Category: games
How should the results be sorted?
 1. by price
 2. by name
Sorting: 1
Sort according to:
 1. Ascending order
 2. Descending order
Ordering: 1
Limit: 2
-- RESULTS --

=== Realm of Battle Skill ===
Product ID: 13
Price: 19.95
Category: games
Description: An MMORPG
```

```
=== Star Fights 3 ===
Product ID: 2
Price: 15.39
Category: games
Description: A space simulator
```

Tips: Merk at man må nå holde styr på hvilke "placeholder" man har lagt til i spørringen og ikke (avhengig av brukerens input). I Python kan det derfor være lurt å bruke navngitte "placeholders" som beskrevet i ukens videoer. For Java blir dette litt mer komplisert, siden Javas JDBC ikke støtter navngitte parametre. Her kan man derimot bruke en teller (f.eks. `int count=0`) og så øke denne for hver "placeholder" man substituerer inne en verdi for. F.eks. som i koden her:

```
String q = "SELECT * FROM ws.products WHERE name LIKE ?";
if (min_price != null)
    q += " AND price <= ?";
if (max_price != null)
    q += " AND price >= ?";
q += ";";

PreparedStatement stmt = connection.prepareStatement(q);
stmt.setString(1, name);
int count = 2;
if (min_price != null)
    stmt.setFloat(count++, min_price);
if (max_price != null)
    stmt.setFloat(count++, max_price);
```

Oppgave 2 – Bestilling av produkter

I denne oppgaven skal du implementere `order_products(conn, username)` (Python) eller `orderProducts(Connection connection, String username)` (Java). Denne metoden blir kalt etter at en bruker har søkt etter produkter. Denne metoden skal spørre brukeren om hvilket produkt brukeren ønsker å bestille basert på `pid` (produktets ID), og deretter hvor mange av dette produktet som skal bestilles. Så skal metoden kjøre en `INSERT`-kommando som setter bestillingen inn i `ws.orders`-tabellen.

Merk at man vet brukernavnet til brukeren som skal bestille produktet, men i `ws.orders` skal man legge in brukerens `uid`.

En bestilling kan se slik ut (og er en forsettelse av søket vist over), hvor brukeren bestiller 5 eksemplarer av spillet med `pid` lik 2, altså `Star Fights 3`:

```
[...]
-- RESULTS --

=== Realm of Battle Skill ===
Product ID: 13
Price: 19.95
Category: games
Description: An MMORPG
```

=== Star Fights 3 ===
Product ID: 2
Price: 15.39
Category: games
Description: A space simulator

Order (Product ID): 2
How many: 5
Product(s) ordered.

Lykke til!