

IN2090 – Databaser og datamodellering

13 – Programmering med SQL

Leif Harald Karlsen
leifhka@ifi.uio.no



Universitetet i Oslo

- ◆ Som oftest er det ikke mennesker som manuelt skriver SQL
- ◆ Men programmer som genererer spørringer som de sender til databasen
- ◆ Spørringene kan da genereres basert på bruker-input, hendelser, el.
- ◆ Naturlig indeling av frontend og backend:
 - ◆ Frontend håndterer input fra bruker, visualiserer av resultater, osv.
 - ◆ Backend svarer på spørringer, utfører kompliserte beregninger, osv.

Eksempel

Går inn på <http://finn.no>'s "Bolig til salgs" og setter:

- ◆ Sted: Oslo eller Akershus
- ◆ Makspris: 5,000,000,-
- ◆ Minste pris: 3,000,000,-
- ◆ Antall rom: 3

og klikker "Søk"

Generert (mulig) SQL-spørring:

```
SELECT *  
  FROM boliger  
 WHERE (sted = 'Oslo'  
        OR sted = 'Akershus')  
 AND pris <= 5000000  
 AND pris >= 3000000  
 AND ant_rom >= 3;
```

Generelle prinsipper

- ◆ Programmer håndterer SQL-spørringer som strenger
- ◆ Kan dermed manipulere SQL-spørringer akkurat som strenger
- ◆ For å kunne sende en spørring til en database trenger man to ting:
 - ◆ En tilkobling – `Connection`
 - ◆ En eller flere spørrings-eksekverere – `Cursor/Statement`

Connection

- ◆ `Connection`-objekter er ansvarlige for å lage en tilkobling til databasen
- ◆ Input til disse er databasenavn, brukernavn, passord, port, osv.
- ◆ Når tilkoblingen er vellykket kan man begynne å lage spørrings-eksekverere fra en `Connection`

Spørrings-eksekverere

- ◆ Lages fra en `Connection`
- ◆ Gis en spørring som en streng
- ◆ Kan så eksekvere spørringen via et metode-kall (typisk `execute()`)
- ◆ Kan så hente ut svarene fra spørringen

Python og Psycopg2

- ◆ Biblioteket for intraksjon med PostgreSQL fra Python heter `psycopg`¹
- ◆ Man starter med å lage et `Connection`-objekt²
- ◆ Fra dette lager man så `Cursor`-objekter³ som kan eksekvere spørringer
- ◆ Spørringene kan så hentes ut som vanlige Python-lister av tupler ved å kalle `cursor.fetchall()`

¹<http://initd.org/psycopg/docs/>

²<http://initd.org/psycopg/docs/connection.html>

³<http://initd.org/psycopg/docs/cursor.html>

Stort eksempel: Webshop

- ◆ Vi skal lage programmer for en nettbutikk
- ◆ Skal i dag lage program som lar brukere
 - ◆ Registrere ny bruker
 - ◆ Logge inn
 - ◆ Søke etter produkter
 - ◆ Bestille produkter
- ◆ I obligen skal dere lage to programmer
 - ◆ Ett som lar ansatte legge inn nye kategorier og produkter
 - ◆ Ett som lager regninger for kunder
- ◆ Viser både Python og Java
- ◆ For obligen velger dere enten Python eller Java

Java og JDBC

- ◆ Biblioteket for interaksjon med databaser fra Java heter JDBC
- ◆ Egen driver for PostgreSQL som lastes inn med `Class.forName("org.postgresql.Driver")`
- ◆ Kan lage `Connection`-objekt⁴-objekt ved å kalle `DriverManager.getConnection(<conStr>)` hvor `<conStr>` er en streng som inneholder en URI med tilkoblingsdetaljer
- ◆ Kan så lage `Statement`⁵/`PreparedStatement`⁶-objekter ved å kalle `connection.createStatement()` eller `connection.prepareStatement()`
- ◆ En spørring eksekveres ved å kalle `statement.execute()`
- ◆ Resultatene fra en spørring kommer i form av et `ResultSet`⁷

⁴<https://docs.oracle.com/javase/8/docs/api/java/sql/Connection.html>

⁵<https://docs.oracle.com/javase/8/docs/api/java/sql/Statement.html>

⁶<https://docs.oracle.com/javase/8/docs/api/java/sql/PreparedStatement.html>

⁷<https://docs.oracle.com/javase/8/docs/api/java/sql/ResultSet.html>

ResultSet

- ◆ Et `ResultSet` holder alltid en peker til én rad i resultatet
- ◆ Man kan hoppe videre til neste rad ved å kalle metoden `next()`
- ◆ Denne metoden returnerer en boolsk verdi som er usann dersom det ikke finnes flere rader i resultatet
- ◆ For hver mulige type har man en egen `get`-metode (f.eks. `getString()`, `getInt()`) som tar en `int` som argument som er kolonne-nummeret
- ◆ Så `result.getString(2)` henter ut verdien i kolonne 2 i den nåværende raden, som en streng