IN2090 – Databaser og datamodellering

11 – Programmering med SQL

Leif Harald Karlsen
leifhka@ifi.uio.no

Universitetet i Oslo

Programmering med databaser

- 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 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, host, osv.
- Når tilkobligen 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`-objekt[2] ved å kalle `psycopg2.connect(connection)`
- Fra dette lager man så `Cursor`-objekter[3] (via `Connection`s `cursor()`-metode)
- `Cursor`-objetket kan så ekvekvere spørringer via `execute(query)` hvor `query` er en streng som inneholder en SQL-spørring
- 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`

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`-objekt[4]-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

Takk for nå!

Neste video introduserer eksempelet vi skal bruke videre.