# JDBC og databaser



# Opgave: forbind Javakode og db sportsclub

- Se README på github
- Lav "Get started"
- Skitser et klassediagram over koden
- Tal med sidemanden om hvordan koden er struktureret
- Vi tager en fælles gennemgang af koden



## SQL fra Java

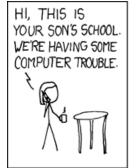
- Med JDBC blander vi to sprog; SQL og Java
- Kompileren kan kun tjekke Java og opfatter SQL som ren tekst – der kan stå hvad som helst
- Hvis vi vil have vores SQL tjekket, kan vi derfor starte i pgAdmin
- Senere kan vi så tilføje vores SQL statements til Javakoden og køre forespørgslerne programmatisk.
- Til sidst kan vi lægge noget lækkert front-end på og så har vi et fuldt system ©

### Statement og PreparedStatement

• Statement er ren SQL:

 PreparedStatement lader os give argumenter med (og er mere sikkert fordi vi undgår SQL injection):

```
String sql = "delete from member where member_id = ?";
ps.setInt( parameterIndex: 1, memberId);
int rowsAffected = ps.executeUpdate();
```









```
INSERT INTO Students (name) VALUES ('Robert');
DROP TABLE Students;
--');
```



# Dependency Injection

- Smart når vi skal lave integrationstest
- Også smart hvis vi skifter vores db ud (fx fra MySQL til Postgres)



# Mapping mellem Java og database

- Klasser/objekter i Java kan svare til tabeller/rækker i databasen
- Hvad kan skabe mismatch mellem Java og databasen?
- Hvilke konstruktioner er unikke for Java og hvilke er for databaser?



### Generated keys

```
try (PreparedStatement ps = connection.prepareStatement(sql, Statement.RETURN_GENERATED_KEYS))
```

```
ResultSet idResultset = ps.getGeneratedKeys();
if (idResultset.next()) {
    newId = idResultset.getInt(columnIndex: 1);
    member.setMemberId(newId);
} else {
    member = null;
}
```

Vi beder PreparedStatement give os den genererede nøgle Vi fortæller, at PreparedStatement skal kunne give os de genererede nøgler



## Synkronisere mellem Java og database

- Hvornår skal vi gøre det?
- Hvordan skal vi gøre det?
- Strategier
  - Kig på nettet
  - Spørg ChatGPT
  - Diskuter fordele og ulemper med hinanden
- Undersøg begrebet Transaktion. Hvordan hænger det sammen med synkronisering?

# DTO – Data Transfer Object

- En måde at oversætte mellem ResultSet og resten af vores kode
- Vi bliver ofte nødt til at pakke vores data ind i et objekt for at kunne sende det rundt i programmet
- Hvorfor sender vi ikke bare ResultSet rundt?
- Hvad er mon forskellen på entiteter/objekter og DTO'er? (tænk fx synkronisering her)