FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

Labo frameworks voor serverapplicaties
JDBC

20 november 2025

1 Inleiding

Tijdens dit labo ontwikkelen we een datalaag die het mogelijk maakt om gegevens van een winkel uit een SQL-databank te halen. De applicatie wordt ontwikkeld m.b.v. JDBC.

2 Opzetten van de databank

In dit labo gebruiken we PostgreSQL als databaseserver. Installeer postgresql als die nog niet geïnstalleerd is op jouw computer. Houd het wachtwoord van de superuser en de poort waarop de databank draait bij. (https://www.postgresql.org/download/)

We voegen nu de databank BIRT toe in PostgreSQL via de commandline. Voeg eventueel het pad "C:\Program Files\PostgreSQL\14\bin" (of het pad waar je postgresql installeerde) toe aan de omgevingsvariabele **`PATH`** zodat je niet de lange padnaam moet gebruiken voor het commando `psql`.

Om de tabellen aan te maken en op te vullen zijn er vier scripts voorzien.

- Het script **`create_classicmodels.sql`** maakt de structuur van de databank aan.
- Het script **`create_user_iii.sql`** maakt een gebruiker "iii" met wachtwoord "iiipwd" aan met alle rechten op de aangemaakte tabellen.
- Het script **`create_stored_procedure.sql`** declareert de stored procedure **`get_total`**.
- Het script **`copy_csv.sql`** voegt data toe aan de tabellen.

Om de tabellen op te vullen zijn er CSV-bestanden voorzien. Je vindt alle bestanden in de zip databank.zip.

Aanmaken databank in het consolevenster Voer de volgende opdrachten op de console uit:

```
psql -U postgres -f create_classicmodels.sql
psql -U postgres -f create_user_iii.sql
psql -U postgres -f create_stored_procedure.sql
psql -U postgres -f copy_csv.sql
```

De structuur van de databank vind je terug in de scripts. In Intellij kan je de databank verkennen: tab Database, vervolgens new (+) DataSource, ... Bij de installatie werd ook pgAdmin voorzien. Dat programma kan je eventueel ook gebruiken om de databank te verkennen.

3 Opdracht

Het project JDBCLabo bevat een console-applicatie `ConsoleApp` en een JUnittest `JdbclaboApplicationTests`. In het bestand `pom.xml` zijn de nodige afhankelijkheden al toegevoegd (JDBC, Postgresql en springboot). Verder zijn er al een heleboel interfaces en klassen voorzien. Bekijk de structuur van het project.

De applicatie die uitgewerkt wordt, voldoet aan volgende eisen:

- In de hoorcollege hebben we twee manieren gezien om een connectie-object aan te maken: via de DriverManager of via de DataSource. In dit labo laat je het Spring-framework een DataSource injecteren. Hiervoor moet je de juiste kenmerken instellen in het bestand `application.properties` en de juiste injector voorzien.

  Zorg dat je connectie steeds correct wordt afgesloten.

- Alle nodige SQL-opdrachten komen uit een configuratiebestand. In het hoorcollege werden twee opties gezien om informatie uit zo'n properties-bestand te lezen: met hulp van een ResourceBundle en met dependency injection zonder expliciete resourcesbundles. Maak in dit labo gebruik van springboot om de waarden te injecteren in je klassen. Meer informatie vind je op https://www.baeldung.com/properties-with-spring

- Zorg dat je applicatie beschermd is tegen SQL-injectie.

- Zorg voor een goede foutafhandeling.

Werk de datalaag (de klasse `JDBCDataStorage`) uit zodat het programma uitgevoerd kan worden en alle testen groen kleuren. Sommige opdrachten werk je uit met een `JdbcClient`, andere niet. Dit staat telkens tussen haakjes aangeduid. De datalaag heeft de volgende functionaliteit:

- Een lijst van alle producten ophalen. (met `JdbcClient`)
- Een lijst van alle klanten ophalen. (zonder `JdbcClient`)
- Een lijst van alle orders van een bepaalde klant ophalen. (met `JdbcClient`)
- Het grootste gebruikte klantnummer bepalen. (zonder `JdbcClient`)
- Het grootste gebruikte ordernummer bepalen. (met `JdbcClient`)
- Een nieuw order inclusief orderdetails toevoegen aan de databank. Zorg ervoor dat wanneer het toevoegen van het order met orderdetails mislukt, de databank terugkeert naar de toestand voor het uitvoeren van de opdracht. (zonder `JdbcClient`)
- Een klant toevoegen (zonder `JdbcClient`)
- Een klant aanpassen (met `JdbcClient`)
- Een klant verwijderen (met `JdbcClient`)
- Het bedrag van alle bestellingen van één klant berekenen. (zonder `JdbcClient`, maak gebruik van de stored procedure)

4 Tips

- Merk op dat niet altijd alle eigenschappen van de toe te voegen objecten ingevuld zijn.

- Vergelijk de constructor van de klasse java.sql.Date met de methode getTime() van java.util.Date. Gebruik de klassen DateFormat en SimpleDateFormat indien nodig.