JDBC - DML

- Definicja:
  DML - (ang. ***D**ata **M**anipulation **L**anguage*), język do zarządzania danymi w bazie, polecenia: SELECT, INSERT, UPDATE, DELETE
  Z poziomu Javy najlepiej wykonywać polecenia DDL za pomocą obiektu PreparedStatement
  https://www.w3schools.in/mysql/ddl-dml-dcl/

  SQLInjections - wstrzyknięcie w normalne zapytanie SQL złośliwego kodu, który ma na celu zniszczyć dane albo uzyskać dane których użytkownik nie powinien widzieć. PreparedStatement zapobiega tego typu atakom.
  https://www.w3schools.com/sql/sql_injection.asp
  https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet

- JDBC API:

  java.sql.PreparedStatement
  prekompilowane zapytanie SQL, można wykonywać wiele razy zmieniając parametry “w locie”, bezpieczeństwo, wydajność, wygoda

- Proste zapytanie i pobranie wyników:

```
int id = 11;
String name = “Adam”;
java.sql.Date date = new java.sql.Date(2018, 6, 10);

PreparedStatement statement = connection.prepareStatement(
          "INSERT INTO table1(id, name, date) VALUES(?, ?, ?)"
          ));
          //parameterIndex zaczyna się od 1!
          statement.setInt(1, id);
          statement.setInt(2, courseId);
          statement.setDate(3, date);

          statement.executeUpdate();
     }
```

Zadanie nr 6:

1. W module javadb-starter w pliku ‘database.properties’ zmień nazwę bazy danych na ‘sda_courses’. Stwórz klasę CoursesManager2 i przenieś tam kod z klasy CoursesManager tak żeby zapytania DML wykorzystywały obiekt PreparedStatement. W klasie CoursesManager2 połącz się z bazą lokalną ‘sda_courses’. Dodaj do odpowiednich metod parametry które będą używane w zapytaniach SQL, np.
   `insertStudent(String name,  int courseId, String description).`

2. Przejdź do modułu: ‘library-manager’. Stwórz metodę DatabaseManager.initializeDb2() i przenieś tam kod z metody initializeDb() wykorzystując obiekt PreparedStatement. Sprawdź czy działa usuwając i dodając strukturę bazy danych ‘library’.

3. Przejdź do modułu: ‘javadb-starter’. Potestuj różne ataki SQLInjection poprzez klasę SqlInjectionSample z modułu javadb-starter. Zmień metody w klasie SqlInjectionSample wykorzystując PreparedStatement i sprawdź czy to zapobiega atakom SQLInjections.