## Day 2 – Backend maintainability and security

1. Sql injection,
2. Views,
3. Data Mapper pattern and
4. 3 layer architecture

### Exercise 1 : Sql Injection

Open the project in the github repos demo folder called: SqlInjectionDemo. In netbeans look at the PartMapperTest. In here there Is a test method called: getPartInjection.

1. Change the input parameter to the getPartByName method so that it takes the sql injection statement instead of the goodsearch statement.
    a. What is the result and why?
2. Create a new test in the test class. The test should make a new injection, that can drop the table
    a. Hint: use this link: http://www.herongyang.com/JDBC/JDBC-ODBC-Flat-File-List-Tables.html to see how to check if a table exists in the database

### Exercise 2: Views

1. Use the mailorder database (find it in the demo folder if you don’t have it already). Create a view that can show orders with their orderliness and the employee responsible for the order. The employee should only be shown with name (not the other columns).
2. Create a new user on the mysql server: ‘viewreader’.
3. Make it so that only viewreader can read from the view
    a. Hint: `GRANT SELECT ON database1.view1 TO 'someuser'@'somehost';`
4. Make a Mapper class: OrderMapper in java that can use the new database user and read from the view.

### Exercise 3: Mappers and Facades

1. Building on the previous exercises java code - add another Mapper class. This mapper should provide methods to select and to insert new customers into the mailorder database.
2. Create a Façade class: DBFacade that can implement a java interface: IDBFacade.java, that combines the public methods of both mapper classes.
3. Create a junit test that can test the methods of IDBFacade.