JDBC and SQL basic security considerations

Database Course

What we'll cover

- SQL user access
- SQL injection

SQL user access

- To limit the effect of anyone hacking our JAVA code we need to make sure our database user can only do what we need it to.
- We also protect ourselves from silly mistakes like dropping the wrong table or updating the wrong table.

```
CREATE USER ‘floggit_service’@'localhost' IDENTIFIED BY 'tomtom';
GRANT SELECT ON floggit.* TO 'floggit_service'@'localhost';
GRANT UPDATE ON floggit.departments TO 'floggit_service'@'localhost';
GRANT INSERT ON floggit.staff TO 'floggit_service'@'localhost';
GRANT DELETE ON floggit.staff TO 'floggit_service'@'localhost';
```

ARCTICTIGER

SQL – injection

- Consider the following.
- We have a textbox on a web page where the user can insert a name and search our db.
- The sql on the server looks like this.

```
sql = “SELECT * FROM staff WHERE name LIKE ‘%” + userInput +”’;
```

ARCTICTIGER

- But what happens if the user is evil and types the following in the text box.

‘; DROP TABLE users; #

sql = “SELECT * FROM staff WHERE name LIKE ‘%‘; DROP TABLE users; ‘”;

- Thankfully the last example doesn't work anymore in the majority of frameworks as you are not allowed to execute 2 statements at a time.

- But password hacks are still definitely possible. See here for a JAVA JDBC example and solution.
- https://www.owasp.org/index.php/Preventing_SQL_Injection_in_Java