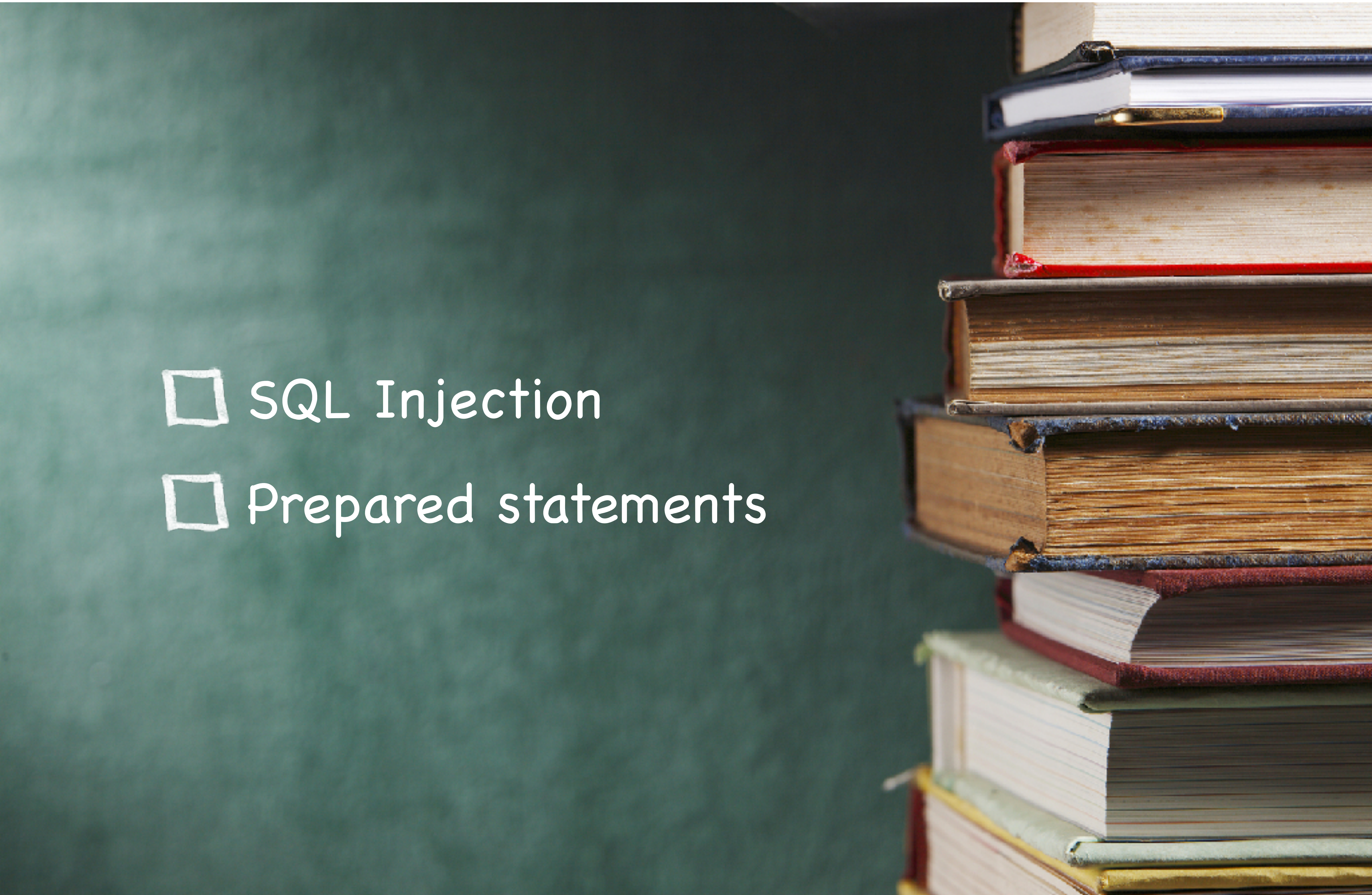
Web 3

Lesson 6: SQL Injection

Agenda



- ☐ SQL Injection
- ☐ Prepared statements

EXAM QUESTIONS…

- ☑ What is SQL injection?
- ☑ Give an example how SQL Injection work?
- ☑ What is the solution we saw for SQL Injection?
- ☑ What are the advantages of prepared statements
- ☑ Name 4 risks we discussed while creating websites.
- ☑ …

OK?

```
public class SqlInjection {

   public static void main(String[] args) throws Exception {
      String email = JOptionPane.showInputDialog(“Enter email");
      String password = JOptionPane.showInputDialog(“Enter password”);
      password = hashPassword(password);

      Connection connection = getDbConnection();
      Statement statement = connection.createStatement();

      String sql = "SELECT * FROM person WHERE email='" + email
           + "' and password='" + password + "'";
      ResultSet result = statement.executeQuery(sql);
      result.next();
      String oldPassword = result.getString("password");

      JOptionPane.showMessageDialog(null, "Your password: " + oldPassword);
   }

   private static String hashPassword(String password) throws Exception {…}

   private static Connection getDbConnection() throws SQLException {…}
}
```

RISK

What if:

```
’ OR 1=1 OR ‘1’=‘1
```

```
...

String sql =
  "SELECT * FROM person WHERE email='" + email
          + "' and password='" + password + “'";

...
```

PROBLEM

- SQL:

```
SELECT * FROM person
   WHERE email=‘’ OR 1=1 OR ‘1’=‘1’;
```

always `true`!

- result:
  - all users en passwords
  - 1st row: probably admin

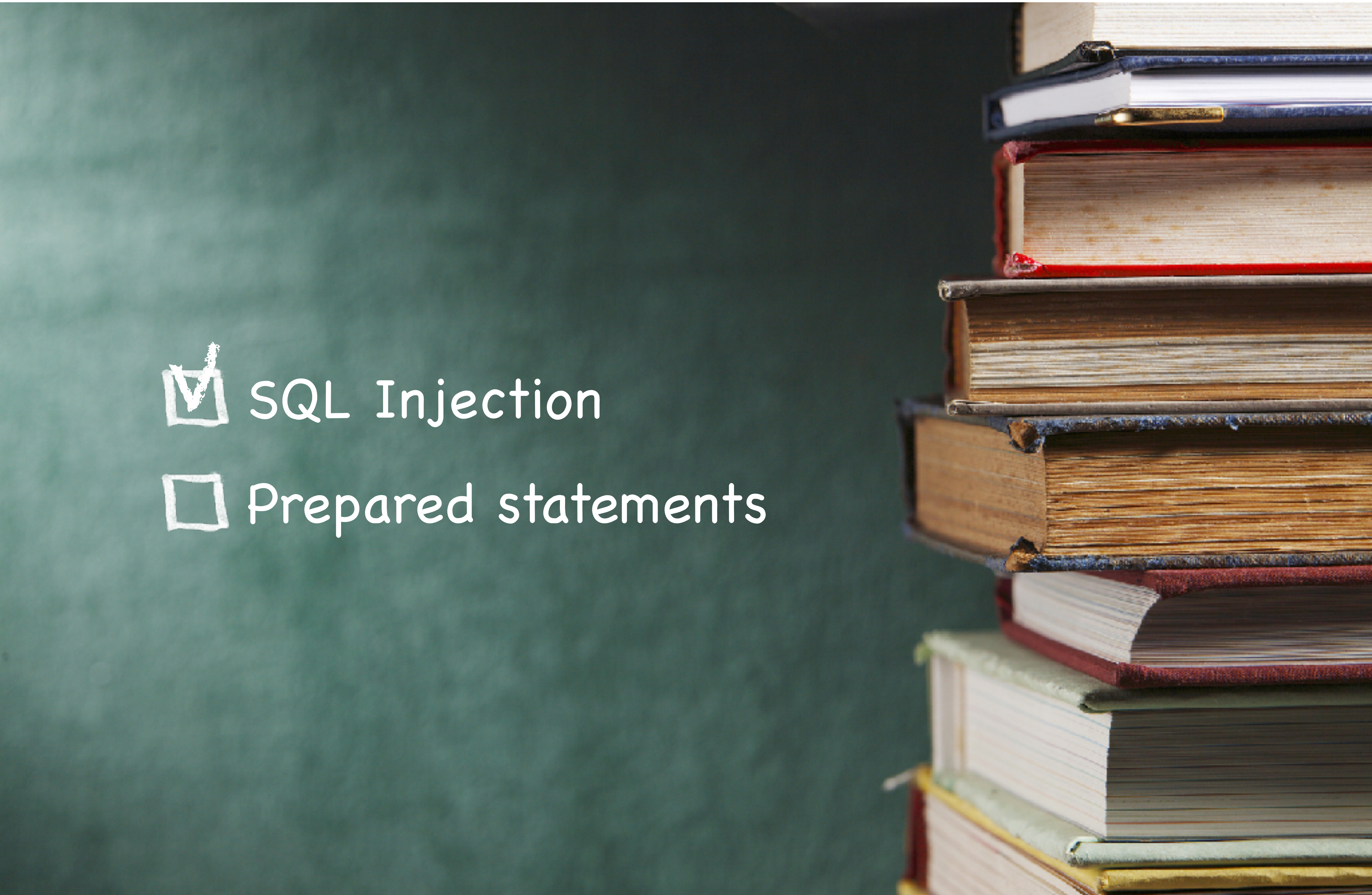
SQL Injection

Code injection in data-driven applications:

- insert malicious SQL statements
- into field
- to get sensitive data or modify database

Solution: **Prepared Statements**

AGENDA



- [✓] SQL Injection
- [ ] Prepared statements

Example

```
public static void main(String[] args) throws Exception {
    // query with a een sql parameter
    String sql = "SELECT * FROM person WHERE email = ?
        and password = ?”;

    // statement is parsed in advance
    PreparedStatement statement = connection.prepareStatement(sql);

    // link Java variables to sql parameters
    statement.setString(1, email);
    statement.setString(2, oldPassword);

    // execute statement
    ResultSet result = statement.executeQuery();
    ...
}
```

PREPARED STATEMENTS

- **Placeholders** instead of values
  - parse statement once
  - call multiple times with other parameter value
- Advantages:
  - **faster**
  - clearer **syntax**
  - structure query is fixed, so s**afe**

REFACTOR USERDB

ADD

```
public void add(Person person) {

   String sql = "INSERT INTO person (name, email, password)"
        + " VALUES (?,?,?)”;

   try (
     Connection connection = DriverManager.getConnection(url, properties);
     Statement statement = connection.prepareStatement(sql);
    ) {

      statement.setString(1, person.getName());
      statement.setString(2, person.getEmail());
      statement.setString(3, person.getPassword());

      statement.execute();
   } catch (SQLException e) {
      throw new DbException(e);
   }
}  ...
```

Refactor UserDb

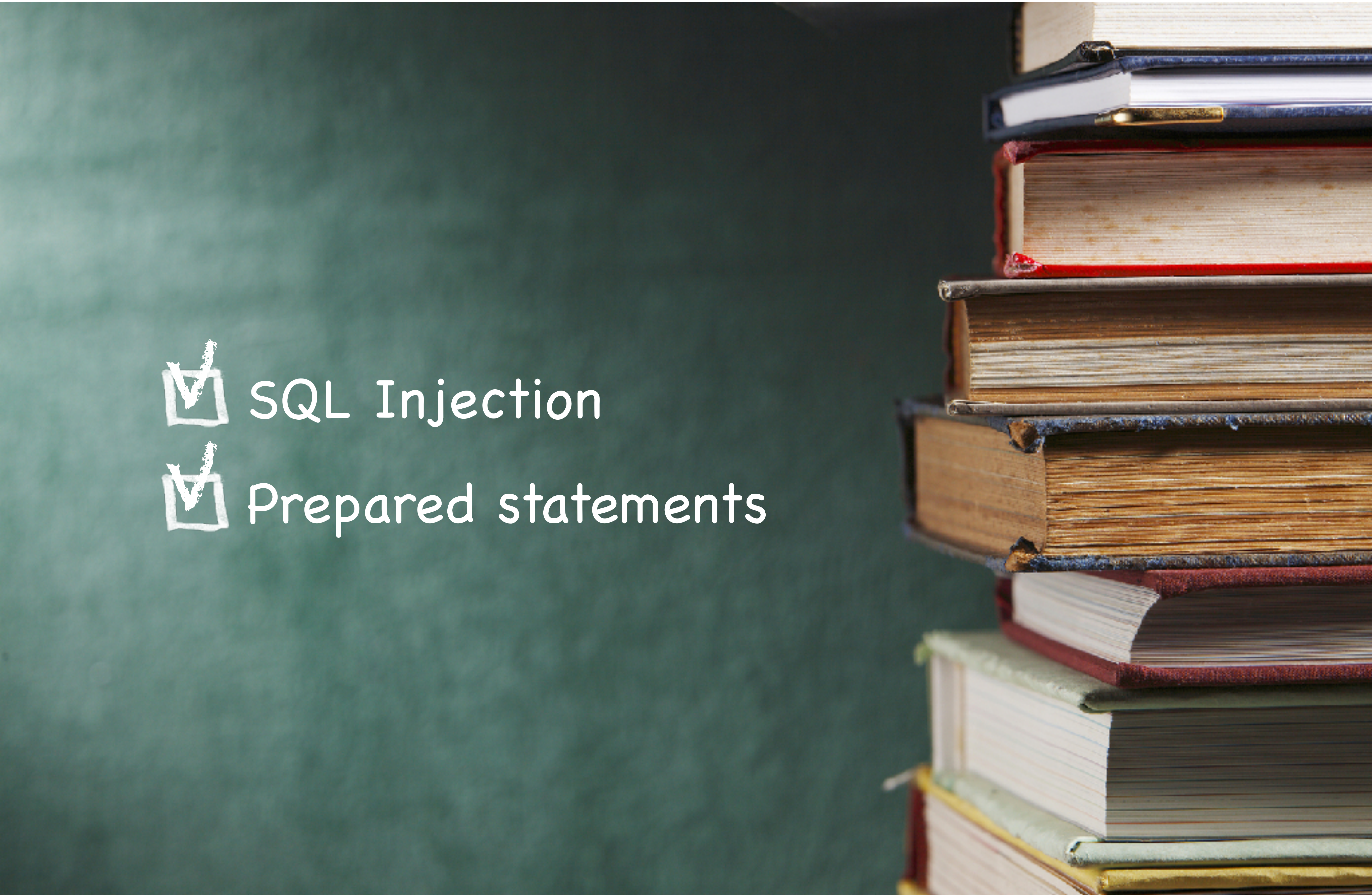
Get

```
public Person get(String email) {
  Person person;
  String sql =
      "SELECT * FROM person WHERE email = ‘" + ?”;
  try (
     Connection connection = DriverManager.getConnection(url, properties);
     Statement statement = connection.prepareStatement(sql);
   ) {
     statement.setString(1, email);
     ResultSet result = statement.executeQuery();
     result.next();
     String name = result.getString("name");
     String password = result.getString("password");
     person = new Person(name, email, password);
  } catch (SQLException e) {
     throw new DbException(e.getMessage(), e);
  }
  return person;
}
```

REDUCING THREADS…

Action	Goal
POST request	Put sensitive parameters in body to hide them
Hashing paswords	Transform passwords to make them unreadable
Output encoding	Replace special characters against Cross-site scripting
Prepared statements	Prepare SQL statements against SQL Injection

Agenda



- ☑ SQL Injection
- ☑ Prepared statements

REFERENCES

- http://www.differencebetween.info/difference-between-encryption-encoding-and-hashing
- https://www.owasp.org/index.php/SQL_Injection