

Object-based Programming

Week 13 – JDBC & Sql2o

Object-Relational Mapping (ORM)

- Lets you query and manipulate data from a database using an object-oriented paradigm
- ORM libraries for Java, e.g. Hibernate, Sql2o

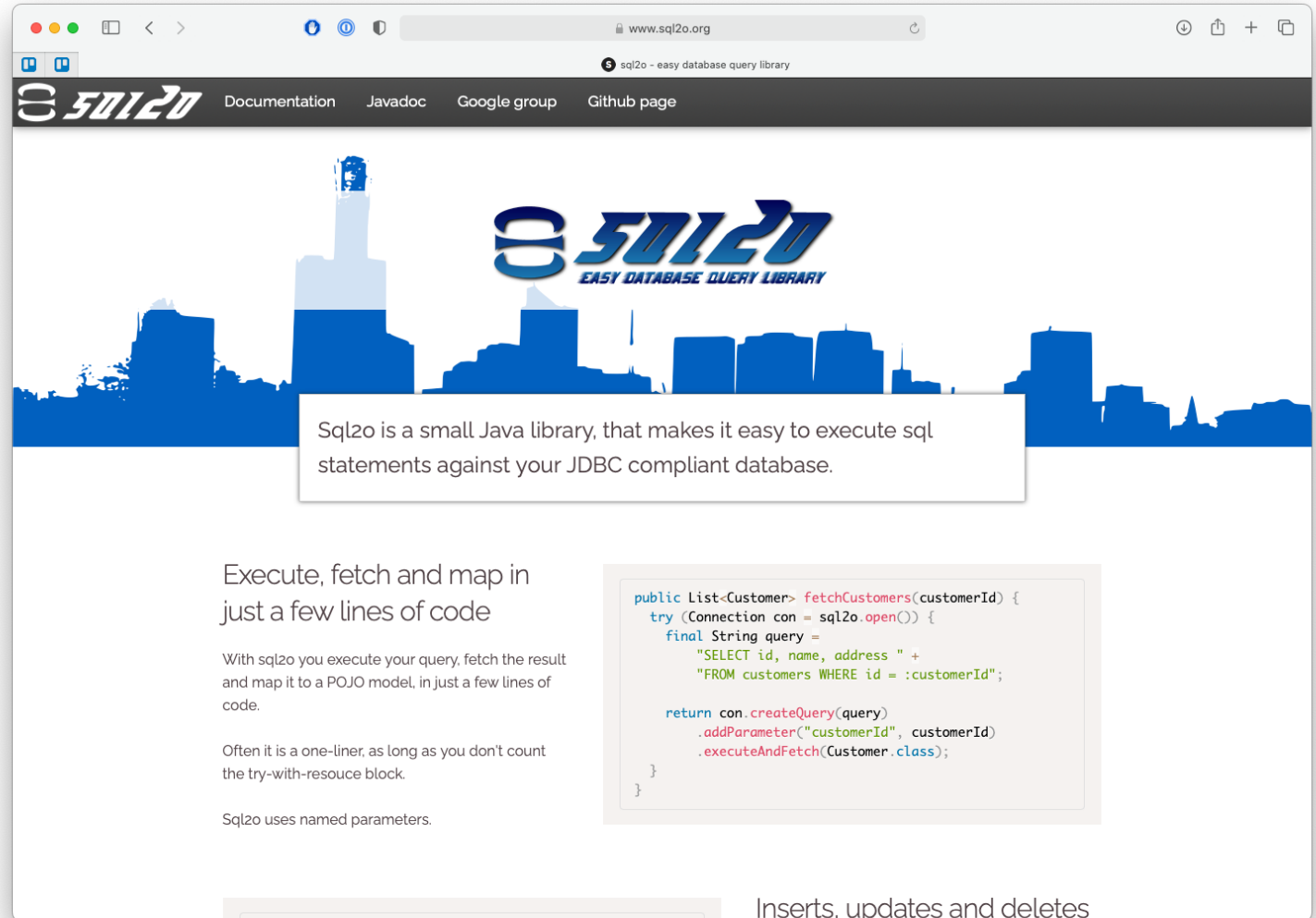


UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Sql2o



The screenshot shows the Sql2o website in a web browser. The browser's address bar displays 'www.sql2o.org'. The website's navigation bar includes links for 'Documentation', 'Javadoc', 'Google group', and 'Github page'. The main header features the 'Sql2o' logo and the tagline 'EASY DATABASE QUERY LIBRARY' above a blue silhouette of a city skyline. A white text box on the skyline reads: 'Sql2o is a small Java library, that makes it easy to execute sql statements against your JDBC compliant database.'

Execute, fetch and map in just a few lines of code

With sql2o you execute your query, fetch the result and map it to a POJO model, in just a few lines of code.

Often it is a one-liner, as long as you don't count the try-with-resource block.

Sql2o uses named parameters.

```
public List<Customer> fetchCustomers(customerId) {  
    try (Connection con = sql2o.open()) {  
        final String query =  
            "SELECT id, name, address " +  
            "FROM customers WHERE id = :customerId";  
  
        return con.createQuery(query)  
            .addParameter("customerId", customerId)  
            .executeAndFetch(Customer.class);  
    }  
}
```

Inserts, updates and deletes

Installation via Maven

- <https://mvnrepository.com/artifact/org.sql2o/sql2o/1.6.0>
- Don't forget to also install the JDBC driver for your DBMS

```
<dependency>  
    <groupId>org.sql2o</groupId>  
    <artifactId>sql2o</artifactId>  
    <version>1.6.0</version>  
</dependency>
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

POJO

Plain Old Java Object

```
public class Employee {  
    private int id;  
    private String fullname;  
    private int salary;  
  
    // getters and setters here  
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Configuration

```
import org.sql2o.Sql2o;

public class Database {
    private static Sql2o db;

    static {
        db = new Sql2o("jdbc:mysql://localhost:3306/CorporateXYZ",
            "username", "password");
    }
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

```
import static Database.db;

public class EmployeeModel {
    ...
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Fetching Data as Models

```
public List<Employee> getAll() {  
    String sql = "SELECT * FROM employee";  
  
    try (Connection con = db.open()) {  
        return con.createQuery(sql)  
            .executeAndFetch(Employee.class);  
    }  
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Fetching Data as Models w/ Parameters

```
public List<Employee> getById(int id) {  
    String sql = "SELECT * FROM employee WHERE id = :id";  
  
    try (Connection con = db.open()) {  
        return con.createQuery(sql)  
            .addParameter("id", id)  
            .executeAndFetch(Employee.class);  
    }  
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Fetching Data as Scalar

```
public Integer getEmployeeCount() {  
    String sql = "SELECT COUNT(id) FROM employee";  
  
    try (Connection con = db.open()) {  
        return con.createQuery(sql)  
            .executeScalar(Integer.class);  
    }  
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Fetching Data as List of Scalars

```
public List<Integer> getEmployeeCount() {  
    String sql = "SELECT id FROM employee";  
  
    try (Connection con = db.open()) {  
        return con.createQuery(sql)  
            .executeScalarList(Integer.class);  
    }  
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

UPDATE

```
public void update(int id, String fullname, int salary) {  
    String sql = "UPDATE employee  
                SET      fullname = :fullname, salary = :salary  
                WHERE   id = :id";  
  
    try (Connection con = db.open()) {  
        con.createQuery(sql)  
            .addParameter("fullname", fullname)  
            .addParameter("salary", salary)  
            .addParameter("id", id)  
            .executeUpdate();  
    }  
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

INSERT

```
public void insert(String fullname, int salary) {  
    String sql = "INSERT INTO employee(fullname, salary)  
                VALUES (:fullname, :salary)";  
  
    try (Connection con = db.open()) {  
        con.createQuery(sql)  
            .addParameter("fullname", fullname)  
            .addParameter("salary", salary)  
            .executeUpdate();  
    }  
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

INSERT and Retrieve id

```
public int insertAndGetId(String fullname, int salary) {  
    String sql = "INSERT INTO employee(fullname, salary)  
        VALUES (:fullname, :salary)";  
  
    try (Connection con = db.open()) {  
        return con.createQuery(sql)  
            .addParameter("fullname", fullname)  
            .addParameter("salary", salary)  
            .executeUpdate()  
            .getKey();  
    }  
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transactions

```
String query1 = "...";
```

```
String query2 = "...";
```

```
try (Connection con = db.beginTransaction()) {  
    con.createQuery(query1).executeUpdate();  
    con.createQuery(query2).executeUpdate();  
    con.commit();  
}
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA