Web Development

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Form Input

Web Development with Play

<input>

```
<form action="/register" method="POST">


      <input type="text" name="firstName">

      <input type="text" name="lastName">

      <input type="text" name="email">

      <input type="password" name="password">


 </form>
```

- Note the ‘name’ attributes

Register Route

```
POST    /register                                  Accounts.register
```

- register() action will be responsible for:
    - Recovering all of the fields “POST”ed by the user
    - Save all these fields in a database
    - Display the start screen again.

```java
public class Accounts extends Controller
{
  //...

  public static void index()
  {
    render();
  }

  public static void register()
  {
    index();
  }

}
```

Controller Parameters

- Controllers can take parameters
- These will be passed from the form
- The names are highly significant

```
public static void register(String firstName, String lastName,
                            String email,     String password)
{
  Logger.info(firstName + " " + lastName + " " + email + " " + password);

  index();
}
```

```html
<form action="/register" method="POST">



      <input type="text" name="firstName">

      <input type="text" name="lastName">

      <input type="text" name="email">

      <input type="text" name="password">



 </form>
```

- Direct mapping from ‘name’ attribute on input element to parameter name in controller/action

```java
public class Accounts extends Controller
{
  //...
  public static void register(String firstName, String lastName,
                              String email,     String password)
  {
    Logger.info(firstName + " " + lastName + " " + email + " " + password);

    index();
  }

}
```

```html
<form action="/register" method="POST">


      <input type="text" name="firstName">

      <input type="text" name="lastName">

      <input type="text" name="email">

      <input type="text" name="password">


 </form>
```

- Direct mapping from ‘name’ attribute on input element to parameter name in controller/action

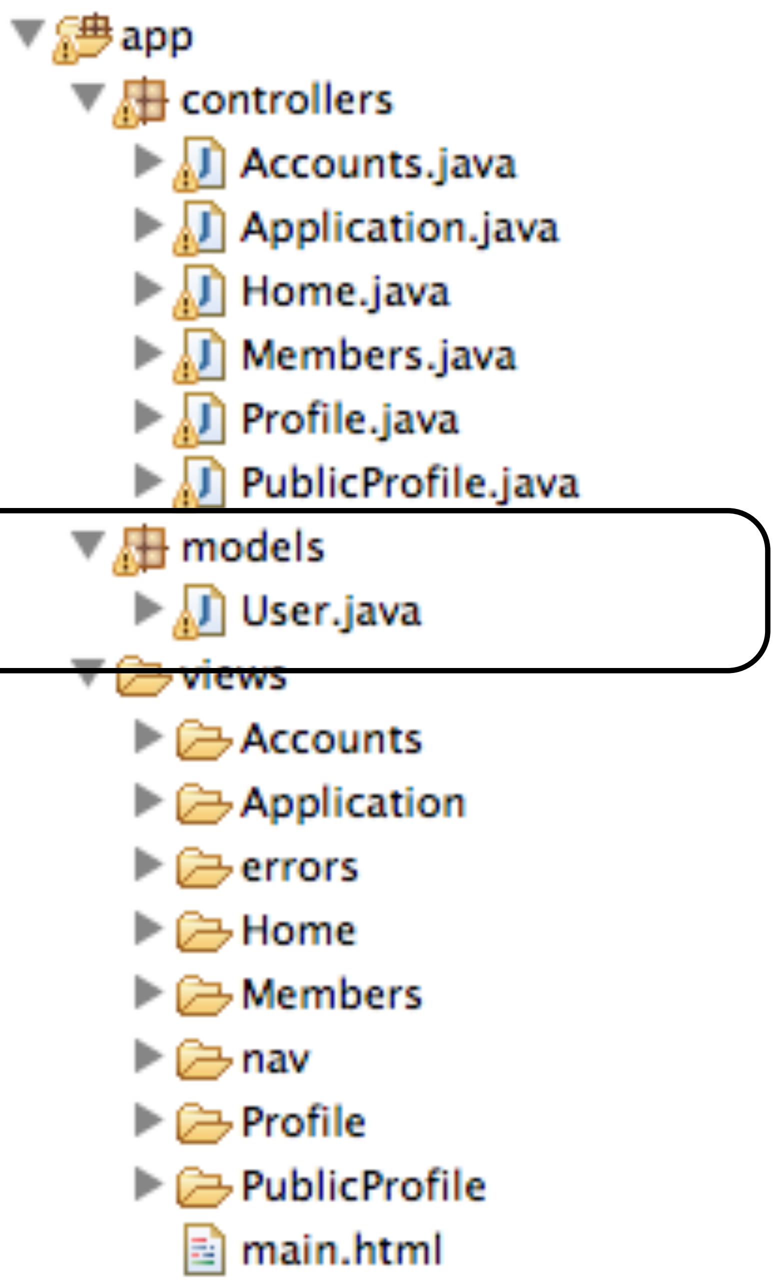
```java
public class Accounts extends Controller
{
  //...
  public static void register(String firstName, String lastName,
                              String email,     String password)

  {
    Logger.info(firstName + " " + lastName + " " + email + " " + password);

    index();
  }

}
```

Database Models



```
▼ app
  ▼ controllers
    ▶ Accounts.java
    ▶ Application.java
    ▶ Home.java
    ▶ Members.java
    ▶ Profile.java
    ▶ PublicProfile.java
  ▼ models
    ▶ User.java
  ▼ views
    ▶ Accounts
    ▶ Application
    ▶ errors
    ▶ Home
    ▶ Members
    ▶ nav
    ▶ Profile
    ▶ PublicProfile
      main.html
```

- We would like to register new users in a database
- In Play, these are represented using 'Models'
- Each table in a database can be represented by a java class
- Instances of this class (objects) will represent rows in the corresponding table

User Model

- Simple class to represent a user
- Public attributes represent fields
- Class ‘extends’ Model and is marked with @Entity annotation to indicate that it is to be saved to a database
- How this is done not our concern

```java
package models;

import javax.persistence.Entity;

import play.db.jpa.Model;

@Entity
public class User extends Model
{
  public String firstName;
  public String lastName;
  public String email;
  public String password;

  public User(String firstName, String lastName,
              String email,     String password)
  {
    this.firstName = firstName;
    this.lastName = lastName;
    this.email = email;
    this.password = password;
  }
}
```

Saving Objects to a Database

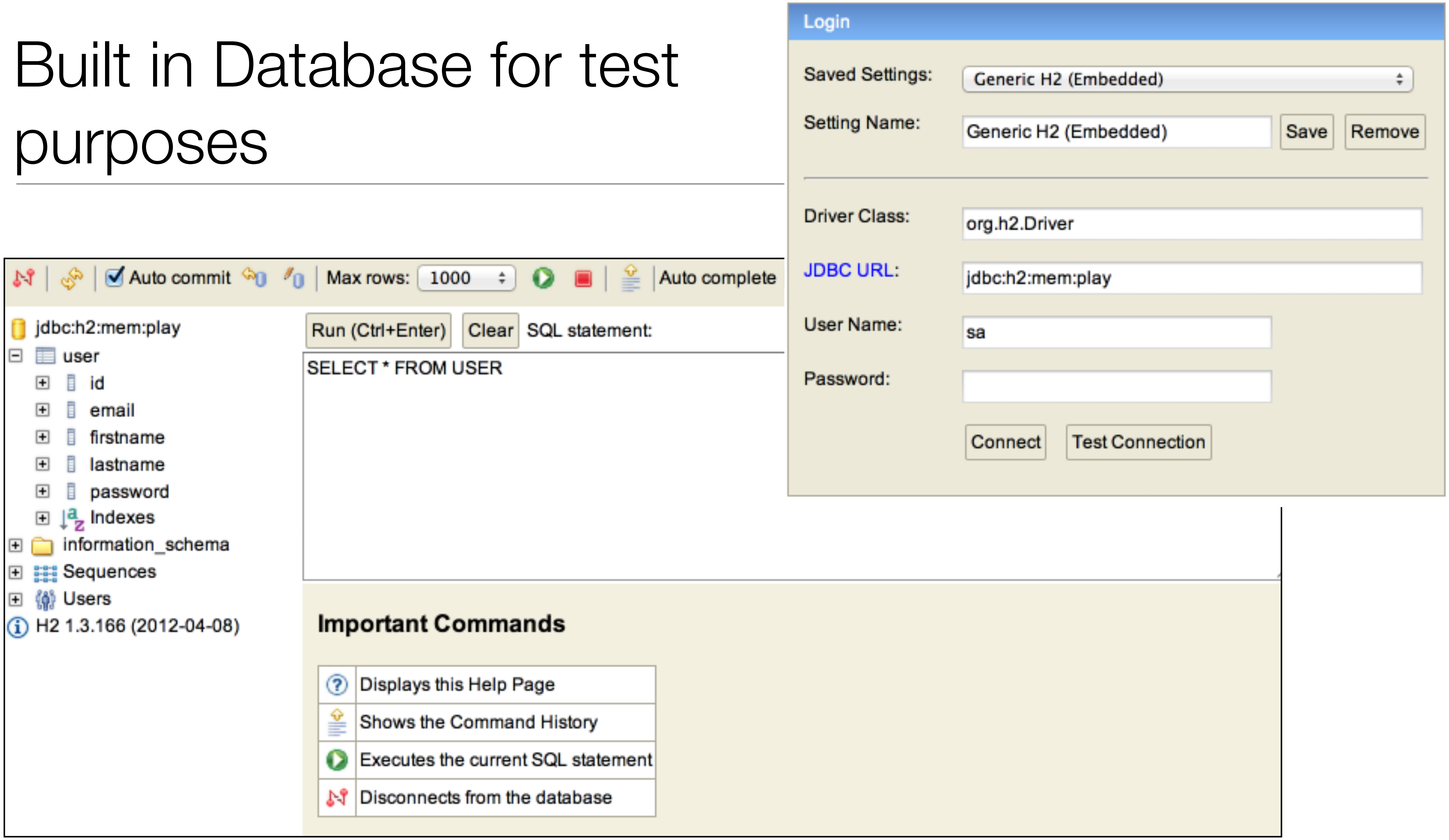
- In register (called when user ‘submits’ signup form):
  - Create a new User object
  - Save it!

```
public static void register(String firstName, String lastName,
                            String email,     String password)
{
  Logger.info(firstName + " " + lastName + " " + email + " " + password);

  User user = new User (firstName, lastName, email, password);
  user.save();

  index();
}
```

Built in Database for test purposes


Login

Saved Settings:	Generic H2 (Embedded)
Setting Name:	Generic H2 (Embedded)
Driver Class:	org.h2.Driver
JDBC URL:	jdbc:h2:mem:play
User Name:	sa
Password:	

```
SELECT * FROM USER
```

jdbc:h2:mem:play
- user
  - id
  - email
  - firstname
  - lastname
  - password
  - Indexes
- information_schema
- Sequences
- Users

H2 1.3.166 (2012-04-08)

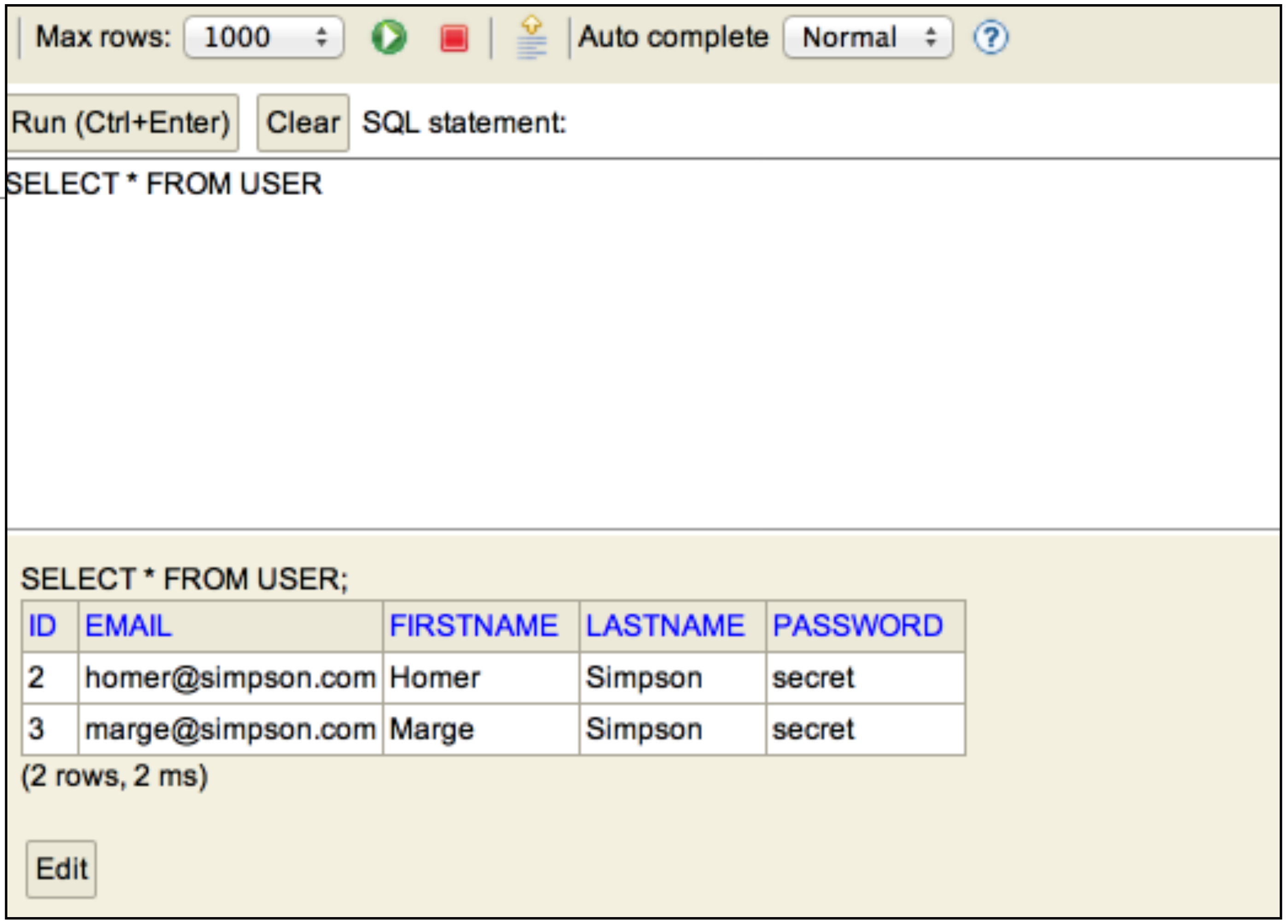
Important Commands

?	Displays this Help Page
	Shows the Command History
▶	Executes the current SQL statement
	Disconnects from the database

- Play comes with a database - which is a full relational db like MySql
- ‘Transient’ - so all values are lost between program executions

Browse/Edit/update...

- Enable in configuration:
    - db=mem
- This means 'in memory' database
- Then just browse to:
    - http://localhost:9000/@db
- when application is running

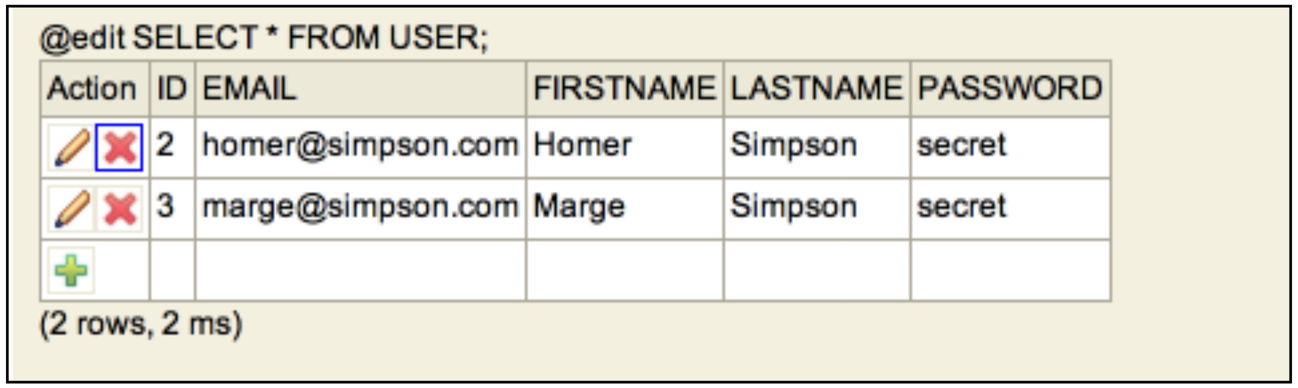


```
SELECT * FROM USER
```

SELECT * FROM USER;

ID	EMAIL	FIRSTNAME	LASTNAME	PASSWORD
2	homer@simpson.com	Homer	Simpson	secret
3	marge@simpson.com	Marge	Simpson	secret

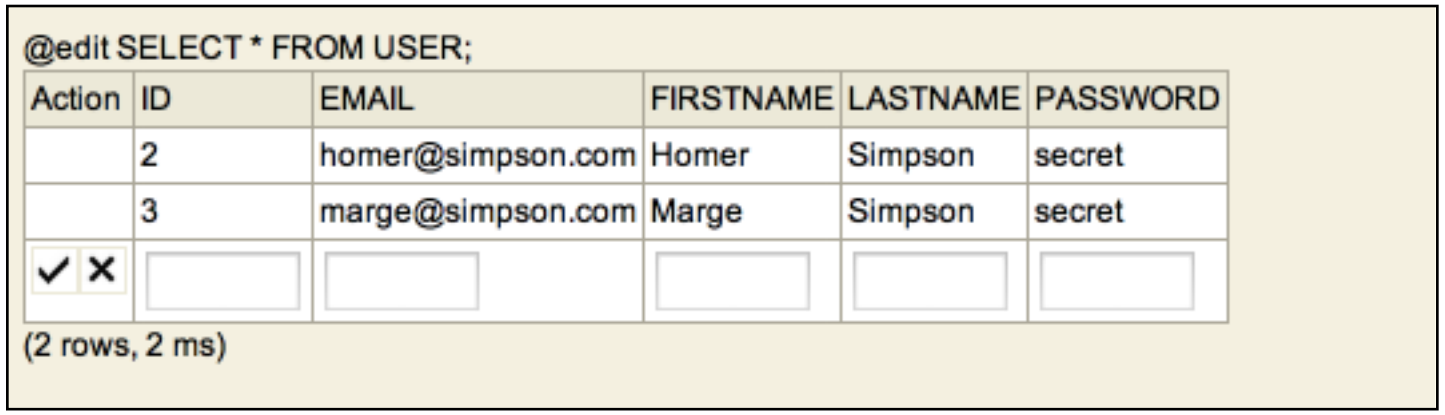
(2 rows, 2 ms)



@edit SELECT * FROM USER;

Action	ID	EMAIL	FIRSTNAME	LASTNAME	PASSWORD
✎ ✗	2	homer@simpson.com	Homer	Simpson	secret
✎ ✗	3	marge@simpson.com	Marge	Simpson	secret
+					

(2 rows, 2 ms)



@edit SELECT * FROM USER;

Action	ID	EMAIL	FIRSTNAME	LASTNAME	PASSWORD
	2	homer@simpson.com	Homer	Simpson	secret
	3	marge@simpson.com	Marge	Simpson	secret
✓ ✗					

(2 rows, 2 ms)

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit