(Paula Prata)

Sessão prática I – Criar JPA "entities" a partir de uma base de dados já existente

1 – Gerar JPA "entities", criar um EJB para aceder à Base de Dados e criar uma Servlet para testar o Bean.

a) Criar uma Web Application (JPAfirst) como descrito anteriormente (Lab 1 ex. 6). Executar a aplicação e observar que a página index.jsp é aberta no Browser definido por omissão. (Se no seu editor foi criada a página index.html, crie a página index.jsp e elimine a primeira.

See Preject. Outcom Project. Prove whether the out of active the set of active the se		Choose Project	Steps	Name and Location	
Web Application Web Control web Ippede Web Control web Application	1. Choose Project 2	Categories: Brojects:	1. Choose Project	Project (jame: JPAfrst	
Web Application Street ad Statigs and the series and Statigs and the series of the series o		Veb Application	 Name and Location Server and Settings 	Project Location: D: WitranJava projetos	Browse
Web Application Street ad Strags Street ad Strags Web Application Web Ap		Javar A Javar A Java Web	4. Frameworks	Protect Folder: D: Witran lava/protetos\/P&first	- Local Andrews
Web Application Image: Control of the solution of project one of the solution of project one of the solution of		- U. Java EE			
When the image is a set of the imag		- 20. HTML/JavaScript		Use Dedicated Folder for Storing Libraries	
When Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupliation librars: Differet care and properts on share the same coupling on properts on propere		- Dava ME		Libraries Folder:	B(oxse
Web Application refrest		- U Maven		Different users and projects can share the same compl	slation libraries
Web Application we protect		- 0 Groovy		gee rep for decaldy.	
Web Application et Project		- 00 C/C++			
Image: Context and project data in a storoid [] project. A storoid project was as: Image: Context and Storoid to built, run, and about project was as: Image: Context and Storoid to built, run, and about project was as: Image: Context and Storoid to built, run, and about project was as: Image: Context and Storoid to built, run, and about project was as: Image: Context and Storoid to built, run, and about project was as: Image: Context and Storoid to built, run, and about project was as: Image: Context and Storoid to built, run, and about project was as: Image: Context and Storoid to built, run, and about project was as: Image: Context and Storoid to built, run, and about project was and Storoid to built. Image: Context and Storoid to built and Dependency Storoid to built.		· · · · ·			
Image: service and settings Image: service and settings <td></td> <td>Qescription:</td> <td></td> <td></td> <td></td>		Qescription:			
Image:		The sentence of the sentence o			
Image: Control Settings Server and Settings Settings <td< td=""><td></td><td></td><td></td><td></td><td></td></td<>					
Image: Control Stations Street and Stations Stations <					
Web Application Server and Settings Server and Settings Server and Settings Add to binaryous Application Server and Settings Best Webs Server and Settings Add to binaryous Application Server and Settings Best Webs Server and Settings Add to binaryous Application Server and Settings Best Webs Server and Settings Deale Official and Dipendency leptons Context fight: (PAdret					
Image: Server and Settings					
Web Application		< Back Next > Brish Cancel Help		< gack Next > Einish	Cancel Help
Add to Stategies and Settings rer and Settings grow: Gaudiata farme grow: Gaudiata fa				E ESULUSUL	
gener: daufah bener · datu an Bit Wenn · find & Arthon · gener / gener / gener · datu an Bit Wenn · find & Arthon · Context Sphi · / PArter	lew Web Application	Server and Settings		resquisur	
June HE Verson Jenes HE Avieté - Estable Contents and Dependency Rejection Content Parts: //PAfris:	lew Web Application	Server and Settings Add to Enterprise Application: @dome> ~		resquisur	
Context Back	ew Web Application ps Choose Project Name and Location Server and Settings Evanaucida	Server and Settings Server and Settings Setting the set of settings Server: Setting the setting to set of setting to setting to setting to setting to set of setting to set	Avu	resquisur	
	ew Web Application s Choose Project Name and Location Server and Settings Frameworks	A Server and Settings did to Enterprise Application: (drune >		Pesquisur	
	lew Web Application ps Choose Project Name and Location Server and Settings Frameworks	Server and Settings	Hall	o World!	
	w Web Application hoose Project ame and Location erver and Settings ameworks		Hell	o World!	
	w Web Application thoose Project time and Location ierver and Settings rameworks	Server and Settings Add to Enterprise Application: (Prove) Prover: Denter (Server) Prover: Prover: Prover: Prove: Pro	Hell	o World!	
	w Web Application	Server and Settings Sold to Enterprise Application: grane :: Generations for the serve grane Ettinescon :: Setting Server Jane Ettinescon :: Setting Server Server :: Setting Server Server :: Setting Server Server :: Setting Server Setting Setting Sett	Hell	o World!	
	w Web Application	Server and Settings got to Sheryne Apolatoks (Sorres	Hell	o World!	
	ew Web Application s Conservation Server and Settings Frameworks	Server and Settings	Hell	o World!	
	lew Web Application ps Couce Project Name and locaton Server and Settings Praneworks	Server and Settings Add to Enterprise Applications: gener: Calantifications: gener: Calantifications: gener: Calantifications: gener: Calantifications: gener: Calantifications: General gene: (PArlies)	Hell	o World!	
	lew Web Application ps Orose Project Orose Project Server and Settings Praneworks	Server and Settings did to Enterprise Application: discuss- gener: Gaterinis force gener: Gaterinis force Gaterinis force Gaterinis and Dependency Injection Context Bath: (JPA-Int)	Hell	o World!	
	ew Web Application ps Choose Priect Imme and Loadon Server and Settings Frameworks	Server and Settings	Hell	o World!	
	ew Web Application p. Crosse Print Server and Settings Praneworks	Add to Enterprise Apolation: (Some)	Hell	o World!	1
	ew Web Application pr Choose Priject Name and Loadon Server and Settings Homework	Server and Settings give :: Gardin Serve	Hell	o World!	1
	wwwebApplication Choose Project Choose Project Server and Settings Frameroots	Evere and Settings Add to Enterprise Apolations: (Preve: dataseting for the set of th	Hell	o World!	
	w Web Application	Server and Settings did to Enterprise Application: (Horse)	Hell	o World!	1

b) Criar uma "entity" para aceder a uma base de dados já existente

- Fazer rigth-click no projeto e selecionar New / Other ... / Persistence / Entity from database /Next

- Selecionar no campo Data Source o valor "jdbc/sample". As tabelas da base de dados exemplo aparecerão em "Available Tables". Selecionar a tabela CUSTOMER e fazer Add.

New File		New Entity Classes from	Database	X
Steps	Choose File Type	Steps	Database Tables	
1. Choose File Type 2	Project: 👹 JPAfirst	1. Choose File Type 2. Database Tables	Data Source: jdbc/sample	•]
	Qategories: Ele Types:	3. Entity Classes	Available Tables:	Sglected Tables:
	Swing GUI Forms Jordeard Objects And TG Life forms Jordeard Objects And TG Life forms Unit Tests Persistence Unit Persistence Unit Descripts form for thirty Classes RESTful Web Services from Entity Classes RESTful Web Services from Entity Classes RESTful Web Services from Entity Classes	 Mapping Options 	MALFACTURER MORCO, MARKT PRODUCT RODUCT_CODE RAICHASE_ORDER (<genove (<genove (<genove (<genove)< th=""><th>DESCOUNT_CODE</th></genove)<></genove </genove </genove 	DESCOUNT_CODE
	Description:			
	Create Java Persstence API entry dasses based on an existing relation disbbase. Entry dasses are used to represent objects shore lides pains in longer than a typical program execution. This template creates an entry dass for each selected table, complete with named query annotations, field representing colume, and relationships representing foreign keys.		Any 🗸	☑ Include Related Tables
	<back next=""> Enish Cance Heb</back>		<gaok next=""></gaok>	Enish Cancel Help

(Paula Prata)

- Após Next introduzir "org.glassfish.samples.entities" para o nome do package. Na coluna Class names aparecem as classes mapeadas da base de dados. Após Next, observar as opções de mapeamento usadas por omissão.

New Entity Classes fro	m Database				×		New Entity Classes from	Database	
Steps	Entity Classe	s					Steps	Mapping Options	i
1. Choose File Type 2. Database Tables	Specify the nar	Specify the names and the location of the entity classes.					1. Choose File Type	Specify the default	mapping options.
3. Entity Classes	<u></u> <u>C</u> lass Names:	Database Table	Class Name	Generation Type			3. Entity Classes	Association Eetch:	default
 Mapping Options 		CUSTOMER	Customer	New			4. Plapping Options	Collection Type:	java.util.Collection
		DISCOUNT_CODE	DiscountCode	New				E Fully Qualified (Database Table Names
								Attributes for Regenerating Tables	
								📝 Use Column Na	mes in Relationships
	Project:	JPAfirst						📃 Uge Defaults if	Possible
	Location:	Source Packages			-			Cenerate Fields	s for Unresolved Relationships
Padgage: arg.glasfish.samples.entities			•						
Generate Named Query Annotations for Persistent Fields									
	💟 Generate 🤰	AXB Annotations							
	Create Pere	iistence Unit							
< Back Next > Emith Cancel [140					P				<back next=""> Finish Cancel Help</back>

- Observar as classes que foram criadas em "Source Packages" "org.glassfisf.samples.entities". Para cada tabela mapeada foi criada uma classe.

As classes geradas usam JPQL (Java Persistence Query Language) para definir queries à base de Dados. Para cada tabela, foram gerados queries para consultar a tabela por cada um dos seus campos e um query para consultar todas as linhas da tabela. Podem ser adicionados novos queries.

São criadas regras de validação "BeanValidation Constraints" para cada campo, baseadas na definição do esquema relacional. Essas restrições serão usadas sempre que uma instância da entity é gravada, atualizada ou removida da base de dados. Restrições baseadas em expressões regulares podem ser usadas para validar os valores. Ver por exemplo o campo phone em Customer.java.

Observando os campos destas duas entidades, DISCOUNT_CODE e CUSTOMER, pode ver-se que existe uma associação (um para vários) em que um desconto está associado a vários clientes e um cliente está associado a um desconto. Esta associação está representada com as anotações @ManyToOne (em Customer) e @OneToMany (em DiscountCode). O campo discountCode permite fazer a junção das tabelas.

```
Em Customer.java
@JoinColumn(name = "DISCOUNT_CODE", referencedColumnName = "DISCOUNT_CODE")
@ManyToOne(optional = false)
private DiscountCode discountCode;
```

Em DiscountCode.java

```
(Paula Prata)
```

```
@Id
@Basic(optional = false)
@NotNull
@Column(name = "DISCOUNT_CODE")
private Character discountCode;
// @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using 1
@Column(name = "RATE")
private BigDecimal rate;
@OneToMany(cascade = CascadeType.ALL, mappedBy = "discountCode")
private Collection<Customer> customerCollection;
```

c) Criar um Bean para fazer queries à base de dados.

- Fazer Right-click em org.glassfish.samples package e criar um Stateless EJB com o nome CustomerSessionBean. Na classe criada, injetar uma instância de EntityManager como ilustrado na figura abaixo. Fazer "Fix Imports".

New Session Bean		
Steps 1. Choose File Type 2. Hame and location	Hame and Location EXB lyame: (Custome/Sessoribean Project: PArtist Location: Source Packages Project: Source Packages Project: Source Packages Project: Source Packages Project: Source Packages Session Type: Source Packages StateSal Source Packages StateSal Source Packages StateSal Source Packages	<pre>@Stateless public class CustomerSessionBean { @PersistenceContext EntityManager em;</pre>

Adicionar à classe CustomerSessionBean o seguinte método:

```
public List<Customer> getCustomer () {
    return (List<Customer>) em.createNamedQuery("Customer.findAll").getResultList();
```

```
}
```

(Atenção: o tipo List pertence ao package java.util)

Este método está a usar o query gerado anteriormente para obter todos os clientes (customers) da base de dados e devolvê-los num objeto do tipo lista de objetos do tipo Customer.

d) Criar uma servlet (de nome TestServlet) para testar o Bean. Observar na Servlet criada a anotação @WebServlet. Para invocar métodos do Bean criado, injetar a classe CustomerSessionBean na Servlet.



No Body da Servlet introduzir a chamada do método getCustomer (). Instrução a bold na listagem abaixo.

```
out.println("<h1>Servlet TestServlet at " + request.getContextPath() + "</h1>");
out.println (ejb.getCustomer());
out.println("</body>");
```

Testar a servlet com rigth-click em TestServlet.java e selecionar Run File. Ou configurar o projeto para iniciar na servlet: Selecionar projeto, em Proprieties / run / relative URL inserir /TestServlet. Ao executar o projeto o Browser mostra a seguinte lista de identificadores:

Project Properties - JPAfint Categories	Server: Java EE Version: Context Path: Context Path: Specify the URI Relative URL: Ception on Sa If selected, file This option file	Gaustriab Server Line IZ 5 hitob (prefet at er in Fun - relative to the critical path to nut; - //restances (or pleansforger, pat) er as complete and digstyped share, you pare them, as possible when you nut a debid your application in the IZE.		[org glasfaå samples entities Customer[customerld=1], org glasfaå samples entities Customer[customerld=2], org glasfaå samples entities Customerl org glasfaå samples entities Customerld=3], org glasfaå samples entities Customerl (customerld=36), org glasfaå samples entities Customerl org glasfaå, samples entities Customerld=31], org glasfaå samples entities Customerl (customerld=36), org glasfaå samples entities Customerl (customerld=31), org glasfaå samples entities Customerl (customerld=36), org glasfaå samples entities Customerl (customerld=31), org glasfaå samples entities Customerld=31), org glasfaå samples entities Customerld=353, org glasfaå samples entities Customerld=310, org gl
		(and for running main classes or unit tasts; e.gtime(2016)	OK Canol 196	org.glastish.smples.entities.Customer[.customerl.d=110]]

2 - Modificar a aplicação para que mostre outros atributos de cada Customer (porque só aparece o ID?).

3 – Modificar a aplicação para executar outros queries.

4 – Mostrar numa servlet os valores da tabela Product.

5 – Explorar como aceder a uma base de dados MySQL.

6 – Crie uma base de dados MySQL exemplo, e repita o processo anterior para essa base de dados.