**Exercícios – JPA**

1) Abra e siga o tutorial de criação de aplicação WEB com JPA no arquivo “tutorial\_jpa”.

2) Ao executar a aplicação do tutorial, responda:

a) Qual(quais) o(s) comando(s) SQL foi(foram) gerado(s) ao consultar a lista de entidades “Customer”?

b) A partir do que foi observado, o relacionamento entre “Customer” e “DiscountCode” é do tipo “eager” ou “lazy”?

c) Você concorda com a forma que a consulta foi implementada pelo EntityManager? Existiria alguma forma de otimização? Qual?

d) Reimplemente a consulta de entidades “Customer” utilizando o seguinte código e observe novamente os resultados de logging.

public List<Customer> findCustomerEntitiesJoin() {

CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery(Customer.class);

Root<Customer> clientes = cq.from(Customer.class);

clientes.fetch("discountCode", JoinType.LEFT);

cq.select(clientes);

TypedQuery<Customer> q = em.createQuery(cq);

return q.getResultList();

}

e) Que característica importante sobre o uso do DataTable pode ser observado no comportamento das consultas SQL realizadas? Você considera o comportamento adequado? Por quê?

f) Tal efeito se deve ao comportamento do controlador do JSF, o qual executa o método de consulta de produtos para cada linha do DataTable. Modifique o método de busca de clientes no bean gerenciado para que ele teste a existência ou não de dados prévios de forma semelhante ao código a seguir e teste novamente.

private List<CustomerDTO> clientes;

public List<CustomerDTO> getCustomersV2(){

if (clientes == null) {

clientes = fachadaEjb.getCustomers();

}

return clientes;

}

3) Adicione uma nova página onde seja possível selecionar um tipo de desconto (via um menu do tipo *dropdownlist*) e listar quais clientes possuem aquele tipo de desconto da base de dados utilizada no tutorial. Não esqueça de alterar a camada de persistência caso seja necessária uma nova operação de consulta.