1Z0-898 Q&As

Java EE 6 Java Persistence API Developer Certified Expert

Pass Oracle 1Z0-898 Exam with 100% Guarantee

QUESTION 1

Which statement is true about the @OrderColumn annotation?

A. If mime is not specified, it defaults to the foreign key column.

B. The OrderColumn annotation may be specified only on a relationship.

C. The OrderColumn annotation is always specified on the owning side of a relationship.

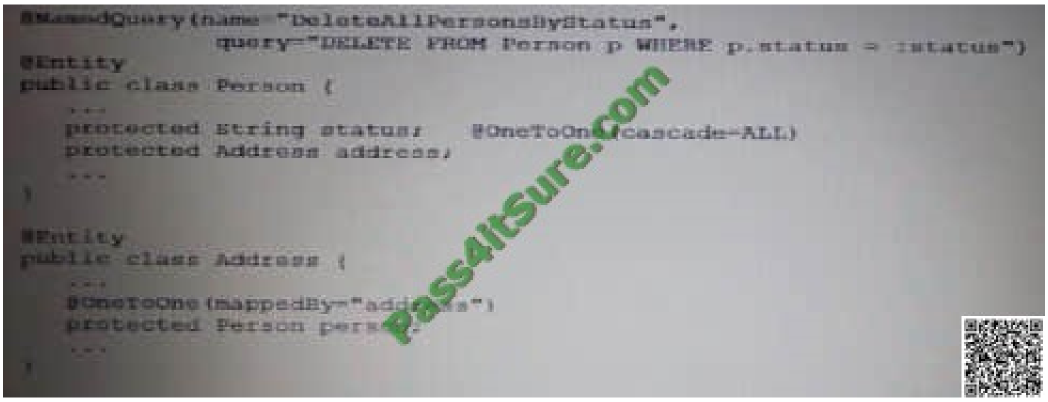
D. The order column is not visible as part of the state of the entity.

Correct Answer: D

Reference:http://www.eclipse.org/eclipselink/api/2.0/javax/persistence/OrderColumn.html(second paragraph)

QUESTION 2

An application has two entities, parson and Address.



```
@NamedQuery(name="DeleteAllPersonsByStatus",
            query="DELETE FROM Person p WHERE p.status = :status")
@Entity
public class Person {
  ...
  protected String status;   @OneToOne(cascade=ALL)
  protected Address address;
  ...
}

@Entity
public class Address {
  ...
  @OneToOne(mappedBy="address")
  protected Person person;
  ...
}
```

The application calls the DeletePersonsByStatus named query. Which of the following is true?

A. The person entities are removed, but NOT their related address entities.

B. The person entities, and all their related address entities, are removed.

C. The DeletePersonsByStatus named query is ill-formed, and will be rejected by the persistence provider.

D. The named query will fall.

Correct Answer: B

QUESTION 3

Given the following code:

```
Public void create () {

try {

doA () {

} catch (PersistenceException e) {}

try (doB) ();

} catch (PersistenceException e) {}

}
```

Calling method doA will cause an NonUniqueResultException to be thrown. Calling method doB will cause an EntityExistsException to be thrown.

What two options describe what will happen when the create method is called within an application \\' uses container managed transactions? (Choose two)

A. Method doB will never be called.

B. The current transaction will continue after doA executes.

C. The current transaction will continue after doB executes.

D. The current transaction will be marked for rollback when doA is called.

E. The current transaction will be marked for rollback when doB is called.

Correct Answer: CE

QUESTION 4

A developer wants to write a type-safe Criteria API query. Which two of the following statements true about Criteria query roots? (Choose two)

A. The query MUST define a query root.

B. The query MUST define a query root only if it navigates to related entities.

C. The query MUST NOT define multiple query roots.

D. The query may define multiple query roots.

Correct Answer: BC

Reference:http://docs.jboss.org/hibernate/orm/4.0/hem/en-US/html/querycriteria.html

http://stackoverflow.com/questions/3424696/jpa-criteria-api-how-to-add-join-clause-as-general- sentence-as-possible

QUESTION 5

The Contact Information embeddable class contains address information as well as a reference to a phone entity. The ContactInformation class is defined as follows: @Embeddable public class ContactInformation

{ String street;

String city;

@OneToOne Phone phone;

}

The developer wants to use this class in an Employee entity, but override the default name of the foreign key to the Phone entity. Which of the code segments shows how to do this correctly?

A. @Entity public class Employee { @Id int empId; @AssociationOverride (name = empInfo.phone", joinColumn = @JoinColumn) (name = "INFO_FK")) ContactInformation empInfo; }

B. @AssociationOverride (name = "empInfo.phone", joinColumn = "INFO_FK") @Id int empId; @ContactInformation empInfo; }

C. @ AssociationOverride (name = "empInfo.phone", joinColumn = "INFO_FK") Entity public class Employee { @Id int empId; }

D. Entity public class Employee { @Id int empId; @ AssociationOverride (name = "empInfo.phone", joinColumn = "INFO_FK") ContactInformation empInfo; }

Correct Answer: A

Reference:http://docs.oracle.com/javaee/6/api/javax/persistence/AssociationOverride.html

QUESTION 6

Which one of the following queries selects the customer whose order has the highest total price?

A. CriteriaBuilder cb = ... Criteria Query cq = cb.create Query (Customer.class); Root c = cq.from(Customer.class); Join o = c.join(Customer .orders); cq.select(c).distinct(true); Subquery sq = cq.subquery(Double.class); Root subo = cq.correlate(o); sq.select(cb.max(subo.get(Order_.totalPrice))); cq.where(cb.equal(o.get(Order_.totalPrice), cb.all(sq)));

B. CriteriaBuilder cb = ... CriteriaQuery cq = cb.createquery(customer.class) Root c = cq.from(Customer.class); Join o = c.join(Customer .orders); cq.select(c).distinct(true); Subquery sq = cq.subquery(Double.class); Root subo = cq.correlate(o); sq.select(cb.max(subo.get(Order_.totalPrice))); cq.where(cb.equal(o.get(Order_.totalPrice), cb.all(sq)));

C. CriteriaBuilder cb = ... CriteriaQuery cq = cb.cteateQuery(Customer.class); Root c = cq.from(Customer.class); Join o = c.join(Customer .orders); cq.select(c).distinct(true); Subquery sq = cq.subquery(Double.class); Root subo = cq.correlate(o); sq.select(cb.max(subo.get(Order_.totalPrice))); cq.where(cb.equal(o.get(Order_.totalPrice), cb.all(sq)));

D. CriteriaBuilder cb = ... CriteriaQuery cq = cb.createQuery(Customer.class); Root c = cq.from(Customer.class); Join o = c.join(Customer_.orders); cq.select(c).distinct(true); Subquery sq = cq.subquery(Double.class); Root subo = sq.from(Order.class); sq. select (ci: . max (subo . get (Order_ . Total Price))) ; cq.where(sq.all(o.gei(Order_.totalPrice)));

Correct Answer: B

QUESTION 7

If an application uses an extended persistence context, which of the following is true?

A. The persistencecontext exists until all transactions invoked by the EntityManager complete.

B. The persistence context exists until all transactions invoked by the EntityManagar complete and the EntityManager.clear () method is invoked.

C. The persistence context exists until the EntityManagerinstance is closed.

D. The persistence context exists until the EntityManagerFactory instance is closed.

Correct Answer: C

Reference:http://java.boot.by/scbcd5-guide/ch07s03.html

QUESTION 8

A session bean business method invokes UserTransaction.setRollbackonly and receives an IllegalStateException. Under which circumstance can this happen?

A. The bean is using bean-managed transactions regardless of whether there is an active transaction.

B. There is no circustance that would cause setRollbackOnly to throw an IllegalStateException.

C. The bean is using bean managed transaction demarcation, and uaerTransaccion.begin has been invoked.

D. The setRollbackOnly method is invoked within a bean-managed transaction, and userTransaction.commit has NOT been invoked.

Correct Answer: C

QUESTION 9

An application uses optimistic locking by defining version attributes in its entity classes. The application performs a bulk update of the entities using a JPQL query. Which of the following is correct?

A. The persistence provider will ensure that the version value in each table is updated.

B. The persistence provider will create a new transaction for the bulk update.

C. An OptimisticLockException will be thrown by the persistence provider.

D. The value of the Version attributes of the updated entitles should be also be explicitly updated by the query.

Correct Answer: A

Reference:http://en.wikibooks.org/wiki/Java_Persistence/Locking(topic: optimistic locking, second paragraph)

QUESTION 10

Consider a persistence application with following entity:

@Entity

Public class MyEntity {

@Column (name = "FIELDA_COLUMN")

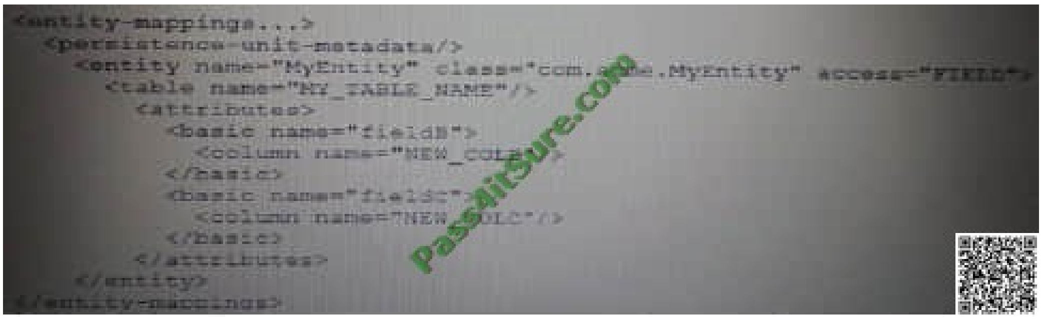
int fieldA;

int fieldB;

int fieldC;

int fieldD;

An orm.xml is packaged in the application with the following contents:



```
<entity-mappings...>
  <persistence-unit-metadata/>
    <entity name="MyEntity" class="com.acme.MyEntity" access="FIELD">
      <table name="MY_TABLE_NAME"/>
        <attributes>
          <basic name="fieldB">
            <column name="NEW_COLB"/>
          </basic>
          <basic name="fieldC">
            <column name="NEW_COLC"/>
          </basic>
        </attributes>
    </entity>
</entity-mappings>
```

Which two of following statement are true. (Choose two)

A. fieldA is mapped to column FIELDA

B. fieldB is mapped to column NEW_COLB

C. fieldD is a persistent attribute of MyEntity

D. fieldD is not a persistence attribute of MyEntity

Correct Answer: BD

QUESTION 11

A developer is writing an application with three java Persistence API entities: order, customer, and Address. There is a many-to-one relationship between order and customer, and a one to-many relationship between customer and Address.

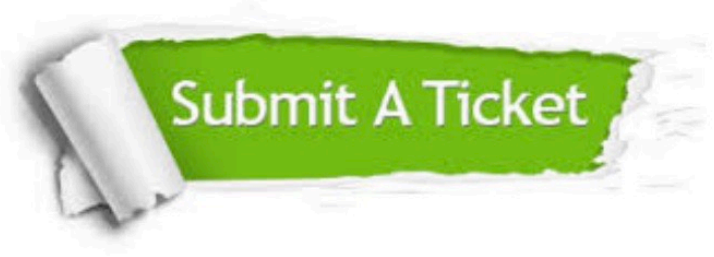
Which two Criteria queries will return the orders of all customers who have an address whose value specified by the String parameter postalcode? (Choose two)

A. String postalCode = . . . Criteria Builder cb = . . . CriteriaQuery cq = cb.createQuery (Order.class); Root order = cq.from(order.class); Join customer = order.join(Order_.customer); Root order = cq.from (Order.class); Join address = customer join (Order_.customer) cq.where (cb.equal (address.get(Address_.postalCode), postalCode)); cq.select (order). Distinct (true); // query execution code here . . .

B. String postalCode = . . . Criteria Builder cb = . . . Root order = cq.from (Order.class); order.join (order_. customer).join(Customer_.addresses); cq.where (cb.equal (address.get(Address_.postalCode), postalCode)); cq.select (order). Distinct (true); // query execution code here

C. String postalCode = ... CritetiaBuilder cb = ... Root order = cq - from (Order . class) , Join address = order.join(Customer_.addresses); cq.where(ct>.equal(address.get(Address_.postalCode), postalCode)); cq-select(order).distinct(true); // query execution code here . . .

D. String postalCode = ... CriteriaBuilder cb = ... Root order = cq- from (Order . class) , Join address = order . join (Order_. customer) -join (Customer_.addresses); cq.where


Submit A Ticket

100%