1Z0-898 Q&As

Java EE 6 Java Persistence API Developer Certified Expert

Pass Oracle 1Z0-898 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/1Z0-898.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Oracle Official Exam Center

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

An entity person is mapped to a table PERSON and has a collection-valued persistence field otherUsedNames that stores names used by a person. The other used Names field is mapped to a separate table called NAMES. Which code fragment correctly defines such field?

A. @ElementCollection (name = "NAMES") Protected set otherUsedNames = new HashSet () ;

B. @Element collection @ElementTable (name = "NAMES") Protected set otherUsedNames = new HashSet () ;

C. @ElementCollection @SecondaryTable (names = "NAMES") Protected set otherUsedNames = new HashSet () ;

D. @ElementCollection @CollectionTable(names = "Names") Protected set otherUsedNames = new HashSet () ;

Correct Answer: D

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

QUESTION 2

A developer has created an application managed entity manager. Which statement is correct?

A. A new persistence context begins when the entity manager is created.

B. A new persistence context begins when a new JTA transaction begins.

C. A new persistence context begins when the entity manager is invoked in the context o\ transaction.

D. A new persistence context begins when the entity manager is invoked in the context of a resource-local transaction.

Correct Answer: B

Reference:http://docs.oracle.com/javaee/6/tutorial/doc/bnbqw.html#bnbra

QUESTION 3

The developer wants to write a criteria query that will return the number of orders made by customer of each county.

Assume that customer is an entity with a unidirectional one-to-many relationship to the Order entity and that Address is an embeddable class, with an attribute country of type String.

Which one of the queries below correctly achieves this?

A. CriteriaBuilder cb> = ... CriteriaQuery cq = cb.createQuery(); Root c = cq.from(Customer.class); Join o = c.join(Customer_.orders); cq.multiselect(cb.count(0), c,get(customer_.address.get(address_.country) cq.groupBy (c.get(customer_.address) .get(address_.country))

B. CriteriaBuilder cb> = ... CriteriaQuery cq = cb.createQuery(); Root c = cq.from(Customer.class); cq.select (cb.count(c.join (customer_. Orders)) , c.get(customers(0), c.get(customer_.address) . get (Address_\\'country)); (c.get(Customer_.address). get(address_.country));

C. CriteriaBuilder cb> = ... CriteriaQuery cq = cb.createQuery(); Root c = cq.from(Customer.class); Join o = c.join(Customer_.orders); cq.select(cb.count(o)); cq.groupBy(c.qet(Customer .address) - get(Address_.country)) ;

D. CriteriaBuilder cb = ... CriteriaQuery cq = cb.createQueryO; Root c = cq.from(Customer.class); Root c = cq . from (Customer . class) ,- Join o = c.join(Customer_.orders); Join country= c.join(Customer,.address) .join(Address cq.multiselect(cq.count(o), country); cq.groupBy(c.get(Customer.address)- get (Address_ . country)) ;

Correct Answer: A

Reference:http://www.jarvana.com/jarvana/view/org/apache/openjpa/openjpa-persistence- jdbc/2.0.0/openjpa-persistence-jdbc-2.0.0-testsources.jar!/org/apache/openjpa/persistence/criteria/TestTypesafeCriteria.java?format=ok

QUESTION 4

The developer wants to define a unidirectional relationship from the customer entity to the order entity and map this relationship using a foreign key mapping strategy.

Which one of the pairs of classes below correctly achieves this task?

A. @Entity public class Customer { @Id int customerId; @OneToMany @JoinColumn (name = "CUST_ID") Set orders; . . . } @Entity public class order { @Id int orderId; . . . }

B. @Entity public class Customer { @Id int customerId; @OneToMany Set orders; . . . } @Entity @JoinColumn (names = "CUST-ID", referencedColumnName = "customerId") public class order { @Id int order Id; . . . }

C. @Entity public class Customer { @Id int customerId; @OneToMany (JoinColumn = @joinColumn (name = "CUST_ID") Set orders; . . . } @Entity public class order { @Id int orderId; . . . }

D. @ Entity public class Customer { @Id int customerId; @OneToMany (JoinColumn = @JoinColumn (name = "CUST_ID"), table = ""ORDER) Set orders; . . . } @Entity public class order { @Id int orderId; . . . }

Correct Answer: A

QUESTION 5

A developer has created a deep entity class hierarchy with many polymorphic relationships between entitles. Which inheritance strategy, as defined by the inheritanceType enumerated type, will be most performed in this scenario?

A. Single table-per-class-hierarchy (InheritanceType.SINGLE_TABLE)

B. Joined-subclass (inheritanceType. JOINED)

C. Table-per-concrete-class (inheritanceType.TABLE_PER_CLASS)

D. Polymorphic join table (inheritanceType. POLYMORPHIC_JOIN_TABLE)

Correct Answer: C

Latest 1Z0-898 Dumps | 1Z0-898 Study Guide | 1Z0-898 Exam Questions