Oracle

Exam 1z0-898

Java EE 6 Java Persistence API Developer Certified Expert Exam

Verson: Demo

[Total Questions: 10]

Question No : 1

A developer who is designing entity classes to map a legacy database encounters a table called STUDENT_RECORD.

This table has two columns, STUDENT_ID and STUDENT_INFO_ID. The primary key of this table consists of both columns, and there is a unique constraint on each info column.

The STUDENT_ID column is foreign key to the STUDENT table and STUDENT_INFO_ID column is a foreign key to the STUDENT_DAT table.

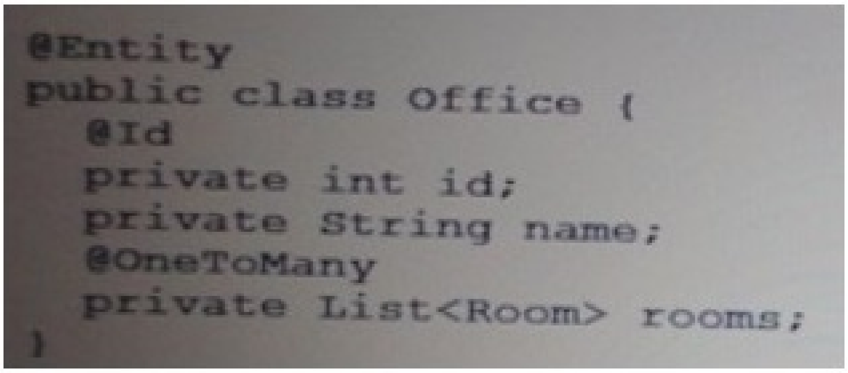
What entity classes and relationships can the developer use to model these tables and relationship?(Choose two)

A. Model the student table as a student entity and STUDENT_DATA table StudentData entity. Model the STUDENT_RECORDS table as bidirectional many-to-many relationship between the student entity on the student data entity.
B. Model the STUDENT table as a Student entity and the STUDENT-DATA table as a StudentData entity. Model the STUDENT_RECORD table as a bidirectional one-to-one relationship between the student entity and the StudentData entity.
C. Model the STUDENT table as a Student entity and the STUDENT-DATA table as a StudentData entity. Model the Student-Records table as a student record entity. Create a many-to-one one relationship from the StudentRecord entity to the student entity and many-to-one relationship from the StudentRecord entity entity to the Student entity and many-to-one relationship from the student entity to the StudentData entity and one-to-many relationship from the StudentData entity to the StudentRecord entity.
D. Model the STUDENT table as a Student entity and the STUDENT-DATA table as a StudentData entity. Model the STUDENT-RECORD table as a StudentRecord entity. Create a bidirectional one-to-one relationship between the StudentRecord entity and bidirectional one-to-one relationship between the Student Record entity and the Student Data entity.

Answer: A,C

Question No : 2

The developer has defined the following entity class office:



```
@Entity
public class Office {
   @Id
   private int id;
   private String name;
   @OneToMany
   private List<Room> rooms;
}
```

Which of the following attributes will be in corresponding generated static metamodel class for the rooms' field?

A. Public static volatile CollectionAttribute<Room> rooms;
B. Public static volatile ListAttribute<Room> rooms;
C. Public static volatile ListAttribute<Office, Room> rooms;
D. Public static volatile SingleAttribute<Room> rooms;

Answer: C

Question No : 3

A developer wrote an entity class with the following method:

Private static Logger logger = Logger.getLogger ("myLogger");

@PrePersist

@PreUpdate

Public void doA () {

Logger.info ("A");

}

@PostPersist

@PostUpdate

Public void doB () {

logger.info ("B");

}

What will the log message contain when an application does the following?

1. Begins a transaction

2. Creates the entity

3. Persists the entity

4. Commits the transaction

5. Begins the entity data

6. Modifies the entity data

7. Merges the entity

8. Commits the second transaction

A. A
A
B
B
B. A
B
A
B
C. A
B
B
A
B
D. The application will throw an exception because multiple lifecycle callback annotations applied to a single method.

Answer: B

Question No : 4

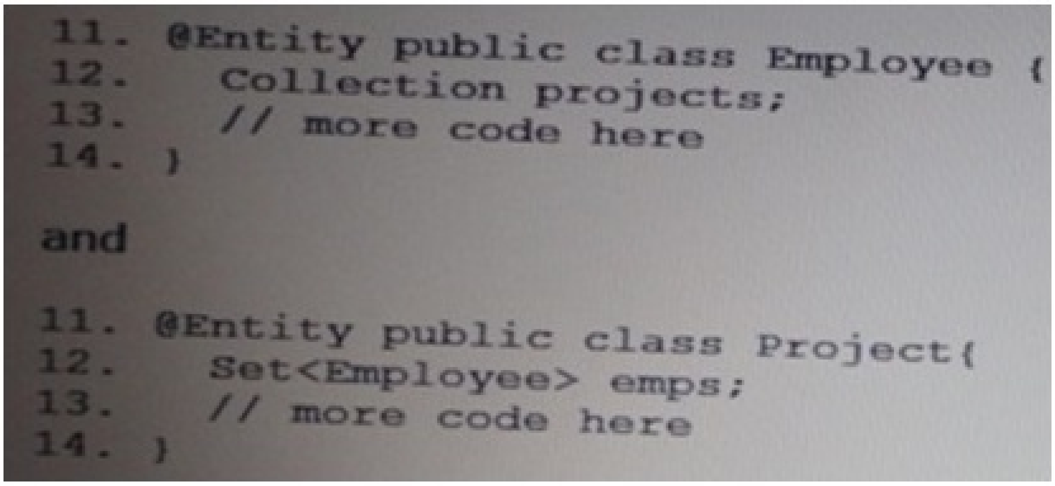
An entity person is mapped to a table PERSON and has a collection-valued persistence field otherUsedNames that stores names used by a person. The other used Names field is mapped to a separate table called NAMES. Which code fragment correctly defines such field?

A. @ElementCollection (name = “NAMES”)
Protected set<String> otherUsedNames = new HashSet () ;
B. @Element collection
@ElementTable (name = “NAMES”)
Protected set<String> otherUsedNames = new HashSet () ;
C. @ElementCollection
@SecondaryTable (names = “NAMES”)
Protected set<String> otherUsedNames = new HashSet () ;
D. @ElementCollection
@CollectionTable (names = “Names”)
Protected set<String> otherUsedNames = new HashSet () ;

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

Question No : 5

Given two entities with many-to-many bidirectional association between them:



```
11. @Entity public class Employee {
12.     Collection projects;
13.     // more code here
14. }
```

and

```
11. @Entity public class Project{
12.     Set<Employee> emps;
13.     // more code here
14. }
```

What set of annotations correctly defines the association?

A. @manyToMany on the projects field,
@manyToMany (mappedBy= “projects”) on the emps field
B. @manyToMany (mappedBy = emps) on the projects field,

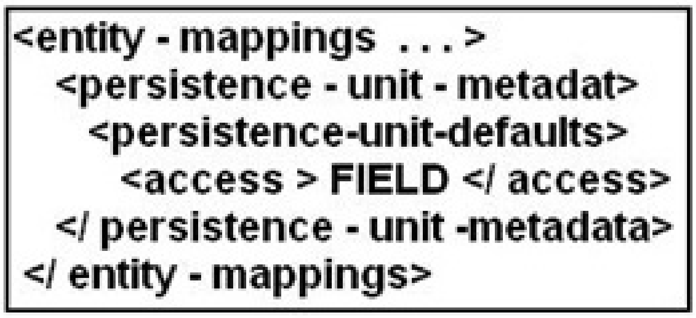
@manyToMany on the emps field
C. @manyToMany ()targetEntity = project.class) on the projects field,
@manyToMany (mappedBy = “projects”) on the emps field
D. @manyToMany (targetEntity = Project.class) on the projects field,
@manyToMany on the emps field

Answer: C

Explanation: http://uaihebert.com/jpa-manytomany-unidirecional-e-bidirecional/

Question No : 6

Consider a persistence application with the following orm.xml:



```
<entity - mappings  . . . >
   <persistence - unit - metadat>
     <persistence-unit-defaults>
       <access > FIELD </ access>
   </ persistence - unit -metadata>
 </ entity - mappings>
```

What will be the effect of the above orm.xml?

A. The access type for only those entities that have not explicitly specified @Access will be defaulted to field.
B. The access type for all entities in the persistence unit will be changed to FIELD.
C. The access type for all entities specified in this orm.xml will be changed to FIELD.
D. The access type for only those entities defined in this orm-xml for which access is not specified will be defaulted to FIELD.

Answer: D

Question No : 7

Given a set of CMT bean methods with the following transaction attributes:

Method M1 = SUPPORTS

Method M2 = REQUIRED

Method M3 = NOT_SUPPORTED

Method M4 = REQUIRES_NEW

And the following method invocation sequence:

Method M1 invokes Method M2

Method M2 invokes Method M3

Method M1 invokes Method M4

If Method M1 is invoked by a method that does NOT have a transaction context, which describes a possible scenario?

A. Method M1:notransaction
MethodM2:newtransaction
MethodM3:notransaction
MethodM4:newtransaction
B. MethodM1:notransaction
MethodM2:ContainerthrowsTransactionNotSupportedException
C. MethodM1:notransaction
MethodM2:runsinsametransactionasM1
MethodM3:containerthrowsTransactionNotSupportException
D. MethodM1:notransaction
MethodM2:newtransaction
MethodM3:ContainerthrowsTransactionNotSupportException.

Answer: A
Explanation: http://docs.oracle.com/javaee/6/api/javax/ejb/TransactionAttributeType.html

QUESTIONNO:34
WhichEntityManagerAPIwilllockentityxwithapessimisticlock?

A.em.lock(x,LockModeType.WRITE)
B.em.lock(x,LockModeType.PESSIMISTIC)
C.em.lock(x,LockModeType.PESSIMISTIC_READ)
D.em.lock(x,LockModeType.OPTIMISTIC_FORCE_INCREMENT)

Answer:C
Reference:http://www.objectdb.com/java/jpa/persistence/lock#Pessimistic_Locking_(pessimisticlocking)

QUESTIONNO:35

ItaPersistenceapplicationlocksentityxwithapessimisticlock,whichstatementistrue?

A.Persistenceprovidercandeferobtainingthelockuntilthenextsynchronizationofanentitytothedatabase

B.APersistenceproviderwillobtainthelockwhentheentityisrefreshedfromthedatabase

C.APersistenceproviderisnotrequiredtosupporttheLockModeType.PESSIMISTIC_WRITElockingtype

D.Ifalockcannotbeobtained,thePersistenceprovidermustthrowanexception

Answer:D

Reference:http://www.eclipse.org/eclipselink/api/2.0/javax/persistence/LockModeType.html

Explanation
:Whenthelockcannotbeobtained,andthedatabaselockingfailureresultsintransaction-levelrollback,theprovidermustthrowthePessimisticLockExceptionandensurethattheJTAtransactionorEntityTransactionhasbeenmarkedforrollback.

QUESTIONNO:36

IfaPersistenceapplicationlocksentityxwithaLockModeType.OPTIMISTIC_FORCEINCREMENTlocktype,whichstatementistrue?

A.ThePersistenceapplicationmustincrementtheversionvaluepriortolockingtheentity.

B.ThisoperationwillresultinaPersistentLockExceptionforanon-versionedobject.

C.ThisoperationwillresultinaPersistentLockExceptioniftheversionchecksfail.

D.LockModeType.OPTIMISTIC_FORCE_INCREMENTisthesynonymoftheLockModeType.
WRITElocktype.

Answer:D

Reference:http://docs.oracle.com/javaee/6/tutorial/doc/gkjiu.html(seventhrowinthefirsttableo nthepage)

QUESTIONNO:37

Persistenceapplicationlocksentityxwithaluck LockModeType.PESSIMISTIC_READlocktype,whichstatementistrue?

A.Thisoperationwillforceserializationamongtransactionsattemptingtoreadtheentitydata.

B.ThisoperationwillresultinaTransactionRolledbackExceptionifthelockcannotbeobtained.

C.Iftheapplicationlaterupdatestheentity,andthechangesareflushedtothedatabase,thelockwill beconvertedtoanexclusivelock.

D.LockModeType.PESSIMISTIC_READisthesynonymoftheLockModeType.READ.

Answer:C

QUESTIONNO:38

Auserentityisretrievedinaqueryandstoredinaninstancevariableuser.Theuserentityhasasingle valuednamepropertythatusesthemappingdefaults,andaphotoproperty,whichislazilyloaded.Theapplicationthencallsthefollowingmethod:

PersistenceUtil.isLoaded(user);

Whichtwoofthefollowingstatementsarecorrect?

A.Thenamepropertywasloadedfromthedatabase.

B.ThenamepropertywasNOTbeloadedfromthedatabase.

C.Thenamepropertymayormaynothavebeenloadedfromthedatabase.

D.Thephotopropertywasloadedfromthedatabase.

E.ThephotopropertywasNOTloadedfromthedatabase.

F.Thephotopropertymayormaynothavebeenloadedfromthedatabase.

Answer:A,F

Reference:http://docs.oracle.com/javaee/6/api/javax/persistence/PersistenceUtil.html http://stackoverflow.com/questions/10437552/what-does-persistenceutil-isloaded-means

QUESTIONNO:39

Giventhefollowingentityclasses:

```
@Entity

@cacheable(true)

PublicclassA{...}

@Entity

@cacheable(false)

PublicclassB{...}

@Entity

PublicclassC{...}
```

Iftheshared-cache-modeelementofpersistence.xmlissettoENABLE_SERVICE,whichentitiesarecachedwhenusingapersistenceproviderthatsupportscaching?

A.Aonly

B.AandB

C.AandC

D.A,B,andC

Answer:A

QUESTIONNO:40

Anapplicationhasthreeentities:themappedsuperclasspersonclassentity,andtheparentandchildentities,whicharesubclassesofperson.

TheapplicationhascreatedfourentityInstances:

Person1isaPersonentitywithaprimarykeyof50

Parent1isaParententitywithaprimarykeyof100

Child1isachiidentitywithaprimarykeyof400

Child2isachildentitywithaprimarykeyof600

Cachinghasbeenenabledinthepersistenceunit,thepersistenceprovidersupportscaching,andnoneofentitieshavetheCacheableannotationapplied,oracacheableXMLelementinpersistence.xml.

Theapplicationexecutesthefollowingcode:

Cachecache=...;

Cache.evict(person.class)

Booleanresult=cache.contains(Child.class,400);

Assumethereisnoconcurrentactivityinvolvingthecache.Whichtwostatementsarecorrect?(Choosetwo)

A.Onlyperson1willberemovedfromcache.

B.Person1,parent1,child1,andchild2willberemovedfromcache.

C.Resultistrue

D.Resultisfalse

Answer:B,D

QUESTIONNO:41

Adeveloperwantstoensurethatanentity'sdataisup-to-datewithregardtothedatabase.Whichofthefollowingstatementsisguaranteedtoaccomplishthis?

A.CallEntityManager.refreshontheentity.

B.Addacacheable(false)annotationontheentityclass.

C.CallEntityManager.findontheentity.

D.Useanamedquerytoretrievetheentity.

Answer:A

QUESTIONNO:42

Anapplicationhastwoentities,DepartmentandEmployee,andthereisaone-to-manyrelationshipbetweenthem.Theapplicationhasthefollowingquery:

SELECTd

FROMDepartmentdLEFTJOINFETCHd.employees

WHEREd.name=:name

Afterreceivingtheresultsofthequery,theapplicationaccessesthereturneddepartment’sEmployeeentitiesstoredintheDepartment.employeescollection-valuedattribute.

Allcachinghasbeenturnedoffintheapplication.

Whichstatementistrue?

A.Thedatabasewillbeaccessedonceduringthequeryexecutionphase,andonceforeachEmployeeentityinDepartment-employees.

B.ThedatabasewillbeaccessedonceduringthequeryexecutionphaseONLY.

C.Thedatabasewillbeaccessedonceduringthequeryexecutionphase,andoncewhenthedepartment.employeescollection-valuedattributeisused.

D.Thedatabasewillbeaccessedonceduringthequeryexecutionphase,oncewhentheDepartment.Employeescollection-valuedattributeisused,andonceforeachemployeeentityintheDepartment.employees.

Answer:B

QUESTIONNO:43

AnapplicationcreatesaTypedQueryobjecttoperformaquery,andsetsthequeryobject’sflushmodebycallingsetFlushMode(FlushModeType.commit).Thequeryisexecutedwithinatransaction.

Whichofthefollowingistrue?

A.Updatestothedatabasetablesmayoccuranytimeduringthetransactionassociatedwiththequery.

B.Updatestotheentitiesthatcanaffecttheoutcomeofthequerycannotbeflushedtothedatabaseuntilthetransactioncommits.

C.Changestotheentitiesinthistransactioncannotbeflushedtothedatabaseuntilthetransactioncommits.

D.setFlushModecannotbecalledonaTypedQueryobject.

Answer:A

QUESTIONNO:44

Adeveloperneedstoincludeasetofmanagedclassesinapersistenceunit.Whichtwosolutionsare correct?(Choosetwo.)

A.Placetheclassfilesintheorm.xmlfile.

B.Placetheclassfilesintherootofthepersistenceunit.

C.Placetheclassfilesinanymappingfilethatisincludedontheclasspath.

D.Placetheclassfilesinanyjarontheclasspaththatisincludedinthepersistenceunit.

Answer:B,D

Reference:https://cwiki.apache.org/GMOxDOC21/persistencexml.html(Topic:<class>)

QUESTIONNO:45

Considerthefollowingwebapplication:

HereMyEntity1.classandMyEntity2.classareannotatedwith@EntityandMyEmbeddable1-classandMyEmbeddable2-classareannotatedwith@Embeddable.MyPUiscontainermanaged.Whichofthefollowingrepresentssetofclassesconsideredmanagedby MyPU?

A.MyEntity1,andMyEmbeddable1

B.MyEntity1,MyEmbeddable1,MyEntity2,andMyEmbeddable2

C.MyEntity1,MyEmbeddable1,andMyEntity2

D.MyEntity1andMyEntity2

Answer:B

QUESTIONNO:46

Considerapersistenceapplicationwithfollowingentity:

@Entity

PublicclassMyEntity{

@Column(name="FIELDA_COLUMN")

intfieldA;

intfieldB;

intfieldC;

intfieldD;

Anorm.xmlispackagedintheapplicationwiththefollowingcontents:

Whichtwooffollowingstatementistrue?(Choosetwo)

A.fieldAismappedtocolumnFIELDA

B.fieldBismappedtocolumnNEW_COLB

C.fieldDisapersistentattributeofMyEntity

D.fieldDisnotapersistenceattributeofMyEntity

Answer:B,C

QUESTIONNO:47

Adeveloperwantstowriteatype-safeCriteriaAPIquery.WhichtwoofthefollowingstatementstrueaboutCriteriaqueryroots?(Choosetwo)

A.ThequeryMUSTdefineaqueryroot.

B.ThequeryMUSTdefineaqueryrootonlyifitnavigatestorelatedentities.

C.ThequeryMUSTNOTdefinemultiplequeryroots.

D.Thequerymaydefinemultiplequeryroots.

Answer:B,D

Reference:http://docs.oracle.com/cd/E19798-01/821-1841/gjitv/index.html

QUESTIONNO:48

WhichofthefollowingCriteriaquerysnippetsdemonstratesthecorrectwaytocreateandexecutestronglytypedqueries?AssumethatcbreferencesaninstanceoftheCriteriaBuilderinterfaceandemreferencesanEntityManagerinstance.

A.CriteriaQuery<office>cq=cb.createQuery(Office.class);

...

TypedQuery<Office>tq=em.ceateQuery(cq);L

1st<office>offices=tq.getResultList();

B.CriteriaQuerycq=cb.createQuery(Office.class)

...

TypedQuery<office>tq=em.createQuery(cq,office.class);

List<office>offices=tq.getresult();

C.CriteriaQuery<office>cq=em.createQuery(cq,office.class);

...

TypedQuery<Office>tq=em.createQuery(cq);

List<office>offices=tq.getresult();

D.CriteriaQuery<office>cq=cb.createQuery(Office.class);

...

TypedQuery<Office>tq=em.ceateQuery(cq);

List<office>Offices=tq.getResultList();

Answer:D

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

QUESTIONNO:49

AdeveloperiswritinganapplicationwiththreejavaPersistenceAPIentities:order,customer,andAddress.Thereisamany-to-onerelationshipbetweenorderandcustomer,andaoneto-manyrelationshipbetweencustomerandAddress.

WhichtwoCriteriaquerieswillreturntheordersofallcustomerswhohaveanaddresswhosevaluespecifiedbytheStringparameterpostalcode?(Choosetwo)

A.StringpostalCode=...

CriteriaBuildercb=...

CriteriaQuery<order>cq=cb.createQuery(Order.class);

Root<order>order=cq.from(order.class);

Join<order,Customer>customer=order.join(Order_.customer);

Root<Order>order=cq.from(Order.class);

Join<customer,Address>address=customerjoin(Order_.customer)

cq.where(cb.equal(address.get(Address_.postalCode),postalCode));

cq.select(order).Distinct(true);

//queryexecutioncodehere

...

B.StringpostalCode=...

CriteriaBuildercb=...

Root<Order>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);

//queryexecutioncodehere

C.StringpostalCode=...

CritetiaBuildercb=...

Root<order>order=cq-from(Order.class),

Join<order,Address>address=order.join(Customer_.addresses);

cq.where(ct>.equal(address.get(Address_.postalCode),postalCode));

cq-select(order).distinct(true);

//queryexecutioncodehere

...

D.StringpostalCode=...

CriteriaBuildercb=...

Root<order>order=cq-from(Order.class),

Join<order,Address>address=order.join(Order_.customer)-join(Customer_.addresses);

cq.where<cb.equal(address.get(Address_.postalCode),postalCode));cq.selec:(order).distinct(true);

//queryexecutioncodehere

...

Answer:A,B

QUESTIONNO:50

Whichoneofthefollowingqueriesselectsthecustomerwhoseorderhasthehighesttotalprice?

A.CriteriaBuildercb=...

CriteriaQuery<Customer>cq=cb.createQuery(Customer.class);

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

Join<Customer,Order>o=c.join(Customer__.orders);

cq.select(c).distinct(true);

Subquery<Double>sq=cq.subquery(Double.class);

Root<Order>subo=cq.correlate(o);

sq.select(cb.max(subo.get(Order_.totalPrice)));

cq.where(cb.equal(o.get(Order_.totalPrice),cb.all(sq)));

B.CriteriaBuildercb=...

CriteriaQuery<Customer>cq=cb.createquery(customer.class)

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

Join<Customer,Order>o=c.join(Customer__.orders);

cq.select(c).distinct(true);

Subquery<Double>sq=cq.subquery(Double.class);

Root<Order>subo=cq.correlate(o);

sq.select(cb.max(subo.get(Order_.totalPrice)));

cq.where(cb.equal(o.get(Order_.totalPrice),cb.all(sq)));

C.CriteriaBuildercb=...

CriteriaQuery<Customer>cq=cb.cteateQuery(Customer.class);

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

Join<Customer,Order>o=c.join(Customer__.orders);

cq.select(c).distinct(true);

Subquery<Double>sq=cq.subquery(Double.class);

Root<Order>subo=cq.correlate(o);

sq.select(cb.max(subo.get(Order_.totalPrice)));

cq.where(cb.equal(o.get(Order_.totalPrice),cb.all(sq)));

D.CriteriaBuildercb=...

CriteriaQuery<Customer>cq=cb.createQuery(Customer.class);

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

Join<Customer,Order>o=c.join(Customer_.orders);

cq.select(c).distinct(true);

Subquery<Double>sq=cq.subquery(Double.class);

Root<Order>subo=sq.from(Order.class);

sq.select(ci:.max(subo.get(Order_.TotalPrice)));

cq.where(sq.all(o.gei(Order_.totalPrice)));

Answer:D

QUESTIONNO:51

Thedeveloperwantstowriteacriteriaquerythatwillreturnthenumberofordersmadebycustomerof eachcounty.

Assumethatcustomerisanentitywithaunidirectionalone-to-manyrelationshiptotheOrderentityandthatAddressisanembeddableclass,withanattributecountryoftypeString.

Whichoneofthequeriesbelowcorrectlyachievesthis?

A.CriteriaBuildercb>=...

CriteriaQuerycq=cb.createQuery();

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

Join<Customer,Order>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.CriteriaBuildercb>=...

CriteriaQuerycq=cb.createQuery();

Root<Customer>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.CriteriaBuildercb>=...

CriteriaQuerycq=cb.createQuery();

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

Join<Customer,Order>o=c.join(Customer_.orders);

cq.select(cb.count(o));

cq.groupBy(c.qet(Customer__.address)-get(Address_.country));

D.CriteriaBuildercb=...

CriteriaQuerycq=cb.createQueryO;

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

Root<Customer>c=cq.from(Customer.class),-

Join<Customer,Order>o=c.join(Customer_.orders);

Join<Address,String>country=c.join(Customer,.address).join(Address

cq.multiselect(cq.count(o),country);

cq.groupBy(c.get(Customer.address)-get(Address_.country));

Answer:A

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

QUESTIONNO:52

ThedeveloperwantstowriteaportablecriteriaquerythatwillordertheordersmadebycustomerJa mesBrownaccordingtoincreasingquantity.Whichoneofthebelowqueriescorrectlyaccomplishe sthattask?

A.CriteriaBuildercb=...

CriteriaQuery<order>cq=cb.createquery<order.class>

Root<customer,order>0=cq.from(customer.class);

Join<customer,order>0=c.Join(customer-.orders);

cq.where(cb.equal(c.get(customer_.name,JamesBrown)));

cq.orderBy(0.get(order_.quantity));

B.

CriteriaBuildercb=...

CriteriaQuery<order>cq=cb.createquery<order.class>

Root<customer,order>0=cq.from(customer.class);

Join<customer,order>0=c.Join(customer-.orders);

cq.where(cb.equal(c.get(customer_.name,JamesBrown)));cq.select(0);

cq.orderBy(0.get(order_.quantity));

C.CriteriaBuildercb=...

CriteriaQuery<order>cq=cb.createquery<order.class>

Root<customer,order>0=cq.from(customer.class);

Join<customer,order>0=c.Join(customer-.orders);

cq.where(cb.equal(c.get(customer_.name,JamesBrown)));

cq.orderBy(0.get(order_.quantity));

cq.select(0);

D.CriteriaBuildercb=...

CriteriaQuery<order>cq=cb.createquery<order.class>

Root<customer,order>0=cq.from(customer.class);

Join<customer,order>0=c.Join(customer-.orders);

cq.where(cb.equal(c.get(customer_.name,JamesBrown)));

cq.orderBy(0.get(order_.quantity));

cq.orderBy("quantity");

Answer:C

QUESTIONNO:53

Anapplicationhastwoentities,parsonandAddress.

TheapplicationcallstheDeletePersonsByStatusnamedquery.

Whichofthefollowingistrue?

A.Thepersonentitiesareremoved,butNOTtheirrelatedaddressentities.

B.Thepersonentities,andalltheirrelatedaddressentities,areremoved.

C.TheDeletePersonsByStatusnamedqueryisill-
formed,andwillberejectedbythepersistenceprovider.

D.Thenamedquerywillfall.

Answer:A

http://stackoverflow.com/questions/4275291/hibernate-doesnt-remove-child-with-named-query-but-removes-with-session-delete

QUESTIONNO:54

Anapplicationusesoptimisticlockingbydefiningversionattributesinitsentityclasses.Theapplicationperformsabulkupdateoftheentitiesusingajpqlquery.

Whichofthefollowingiscorrect?

A.Thepersistenceproviderwillensurethattheversionvalueineachtableisupdated.

B.Thepersistenceproviderwillcreateanewtransactionforthebulkupdate.

C.AnOptimisticLockExceptionwillbethrownbythepersistenceprovider.

D.ThevalueoftheVersionattributesoftheupdatedentitlesshouldbealsobeexplicitlyupdatedbyth equery.

Answer:D

Reference:http://en.wikibooks.org/wiki/Java_Persistence/Locking(topic:optimisticlocking,se condparagraph)

QUESTIONNO:55

AnamedquerythatsetsanexclusivepessimisticontheentitiesreturnedbythequerybysettingtheN amedQuerylockModeelementtoLockModeType.PESSIMISTIC_FORCE_INCREMENT.The applicationstartstransactionandexecutesthequery.

Whichofthefollowingstatementsiscorrectabouttheentitiesreturnedbythequery?

A.Onlythecurrenttransitionmaymodifyordeletetheentityinstances.

B.ThecurrenttransactionmayNOTmodifyordeletetheentityinstances.

C.Otherconcurrenttransactionsmaymodifyordeletetheentityinstances.

D.OtherconcurrenttransactionsmaymodifybutMAYNOTdeletetheentityinstances.

Answer:A

QUESTIONNO:56

AdeveloperwantstocreateaJavaPersistencequerythatwillincludeasubquery.

Whichthreearetrue?(Choosethree.)

A.SubqueriescanbeusedinaFROMclause.

B.SubqueriescanbeusedinaWHEREclause.

C.TheANYexpressioncanbeusedonlywithasubquery.

D.TheEXISTSexpressioncanbeusedonlywithasubquery.

E.TheMEMBERexpressioncanbeusedonlywithasubquery.

Answer:B,C,D

QUESTIONNO:57

WhichstatementiscorrectabouttheJavaPersistenceAPIsupportfortheSQLqueries?

A.SQLqueriesareNOTallowedtouseparameters.

B.TheresultofanSQLqueryisnotlimitedtoentities.

C.OnlySELECTSQLqueriesarerequiredtobesupported.

D.SQLqueriesareexpectedtobeportableacrossdatabases.

Answer:B

QUESTIONNO:58

Auserentityisinaone-to-manyrelationshipwithaBookentity.Inotherwords,adeveloperreachthecollectionofbooksthatauserinstancemyUserhasbyusingthepathexpression-“myuser-books".

AdeveloperwantstowriteaJavaPersistencequerythatreturnsalluserstathaveonlytwobooks.

Whichtwoarevalidqueriesthatreturnthisinformation?(Choosetwo.)

A.SELECTuFROMUserUWHERESIZE(u.books)=2

B.SELECTuFROMUserWHERECOUNT(u.books)=2

C.SELECTuFROMUseru(WHERECOUNT(b)FROMu.booksb)=2

D.SELECTuFROMuseruWHERE(SELECTSIZE(b)FROMu.booksb)=2

Answer:A,C

QUESTIONNO:59

AsessionbeanbusinessmethodinvokesUserTransaction.setRollbackonlyandreceivesanIllegalStateException.

Underwhichcircumstancecanthishappen?

A.Thebeanisusingbean-managedtransactionsregardlessofwhetherthereisanactivetransaction.

B.ThereisnocircustancethatwouldcausesetRollbackOnlytothrowanIllegalStateException.

C.Thebeanisusingbeanmanagedtransactiondemarcation,anduaerTransaccion.beginhasbeeninvoked.

D.ThesetRollbackOnlymethodisinvokedwithinabean-managedtransaction,anduserTransaction.commithasNOTbeeninvoked.

Answer:A

http://docs.oracle.com/javaee/6/api/javax/ejb/EJBContext.html#setRollbackOnly%28%29

QUESTIONNO:60

AJavaEEapplicationispackagedasfollows.

WhichofthefollowingistrueforaportableJavaEEapplication?

A.Thisisaninvalidapplication.AJavaEEapplicationcannothavemorethanonepersistentwithsamename.

B.“MyPu”definedundereachmoduleisvisibletoonlythedefiningmodule.Thereisnowayothermodulescanaccessit.

C.Codeintheejb1.jarcanaccess“MyPU”definedunderwar1.warusing“war1#myPU”

Answer:B

QUESTIONNO:61

Thedeveloperhasmodeledstudentinterestsasaset<String>:

@EntitypublicclassStudent{

@Idintstudentld;

stringname;

@ElementaryCollection

Set<String>Interests;

...

}

Thedeveloperwantsthevaluesofthissettobestoredusingacolumnnamedstudent_intersets.

Selecttheitembelowthataccomplishesthistask:

A.@ElementaryCollection

@Column(name="student_interests")

Set<string>interests;

B.@ElementaryCollection(column="student_intersets")Set<String>interests;

C.@ElementaryCollection@CollectionTable(column="student_intersets")Set<String>interes ts;

D.@ElementaryCollection@CollectionTable(column=@column(name="student_interests")) Set<String>interests;

Answer:A

Reference:http://en.wikibooks.org/wiki/Java_Persistence/ElementCollection(seeExampleof aelementcollectionrelationshiptoabasicvalueannotations)

QUESTIONNO:62

TheembeddableclassContractInformationisusedinanelementcollectionoftheEmployeeentity.

@Entity

PublicclassEmployee{

@IdintempId;

@ElementaryCollectionSet<ContractInformation>info;

...

}

Assumethatthephoneclassisanentityandthataddressisanembeddedclass.

WhichtwoofthecodesegmentsbelowcanbeusedtomodelthestateofContractInformation?(Choosetwo)

A.@OneToManySet<phone>phones;

B.@EmbeddableAddressaddress;

C.@ManyToOnephonephone;

D.@ElementaryCollection<Phone>phones;

E.@OneToOneAddressaddress;

Answer:B,C

QUESTIONNO:63

IfaPersistenceapplicationlocksentityxwithapessimisticlock,whichstatementistrue?

A.ThePersistentproviderwilllockthedatabaserow(s)thatcorrespondtoallpersistentfieldsofpropertiesofaninstance,includingelementcollections.

B.Onlysingletableperclasshierarchymappingissupportedwiththislocktype.

C.APersistenceproviderwilllocktheentityrelationshipsforwhichthelockedentitycontainstheforeignkey.

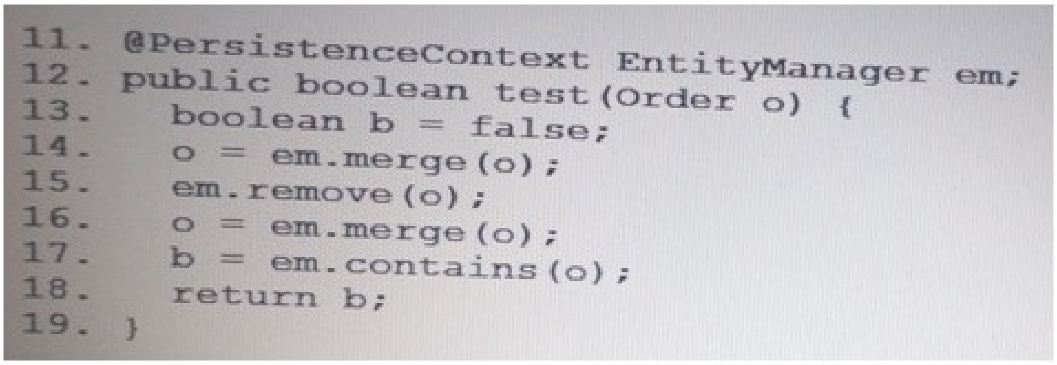
D.Aseparatelockstatementmustbecalledforeachsubclassinentityhierarchy.

Answer:C

Reference:http://docs.oracle.com/javaee/6/api/javax/persistence/PessimisticLockScope.html(searchpublicstaticfinalpessimisticlockscope)

Question No : 8

Given:



```
@PersistenceContext EntityManager em;
public boolean test(Order o) {
   boolean b = false;
   o = em.merge(o);
   em.remove(o);
   o = em.merge(o);
   b = em.contains(o);
   return b;
}
```

Which statement is correct?

A. The method will return TRUE.
B. The method will return FALSE.
C. The method will throw an exception.
D. The order instance will be removed from the database.

Answer: C

Question No : 9

The developer wants to override the default mappings for an embeddable class Address used by the customer entity.

The Address class is defined as follows:

@Embeddable public class Address (

private String street;

private String city;

private String country;

. . .

)

Assume that NO mapping descriptor is present. Which code segment below shows the correct way to override the default mapping for address?

A. @AttributeOverrides ({
@AttributeOverride (name = “street”, column = @Column (name = ADDR_STREET)),
@AttributeOverride (name = “city, column = @Column (name = ADDR_CITY)),
@AttributeOverride (name = “country, column = @Column (name = ADDR_COUNTRY)),
}}
@Embedded Address addr;

B. @ AttributeOverrides ({
@AttributeOverride (name = “street”, column = @Column (name = “name_STREET”)),
@AttributeOverride (name = “city, column = @Column (name = “name_CITY”)),
@AttributeOverride (name = “country, column = @Column (name = “name_COUNTRY”)),
}}
@Embedded Address addr;

C. @ AttributeOverrides ({
@AttributeOverride (name = “street”, column (name = “name_STREET”)),
@AttributeOverride (name = “city, column (name = “name_CITY”)),
@AttributeOverride (name = “country, column (name = “name_COUNTRY”)),
}}
@Embedded Address addr;

D. @AttributeOverrides ({
@AttributeOverride (name = “addr.street”, column = @Column (name = ADDR_STREET)),
@AttributeOverride (name = “addr.city”, column = @Column (name = ADDR_CITY)),
@AttributeOverride (name = “addr.country”, column = @Column (name = ADDR_COUNTRY)),
}}
@Embedded Address addr;

Answer: B

Reference:http://docs.oracle.com/javaee/5/api/javax/persistence/AttributeOverrides.html

Question No : 10

An application wants to utilize side effects of cascading entity manager operations to related entities.

Which statement is correct?

A. The persist operation is always cascaded to related entitles for one-to one and one-to-many relationships.
B. To minimize the effect of the remove operation applied to an entity participating in a many-to-many relationship the remove operation should he cascade to entities on both sides of the relationship.
C. The persist operation applied to a new entity x is cascaded to entities referenced by x if the relationship from x to these other entities is annotated with the cascade=PERSIST or cascade=ALL annotation element value.
D. The remove operation applied to a removed entity x is cascaded to entities referenced by x of the relationship from x to these other entities is annotated with the cascade = REMOVE of cascade = ALL annotation element value.

Answer: C
Reference:http://stackoverflow.com/questions/4748426/cannot-remove-entity-which-is-target-of-onetoone-relation(answer 1)