Comenzado el	miércoles, 20 de octubre de 2021, 22:45
Estado	Finalizado
Finalizado en	miércoles, 20 de octubre de 2021, 23:00
Tiempo empleado	15 minutos 3 segundos
Puntos	17,00/17,00

Pregunta **1**

Correcta

Puntúa 1,00 sobre 1,00

Information Repositories

Questionnaire

What options do you have to support with tables an inheritance hierarchy in which the base class is abstract?

Seleccione una:

- a. Add one table per class.
- b. Add a single table with the columns of all the tables.
- c. Add a table for each non-abstract class.
- d. All are valid, depending on the circumstances. ✔

null

La respuesta correcta es: All are valid, depending on the circumstances.

Pregunta **2**

Correcta

Puntúa 1,00 sobre 1,00

If I have a hierarchy of classes in which the base class has the most attributes and I don't mind having null fields in the tables, which strategy do you think is better?

Seleccione una:

- a. One table per class.
- b. A single table. ✔
- c. One table per class not abstract.
- d. All are equal.

null

La respuesta correcta es: A single table.

Pregunta **3**

Correcta

Puntúa 1,00 sobre 1,00

If I have a hierarchy of classes with two children in which one has many columns but few rows and the other class has few columns and many rows, which strategy do you think is worse?

Seleccione una:

- a. One table per class.
- b. A single table. ✔
- c. One table per class not abstract.
- d. All are equal.

null

La respuesta correcta es: A single table.

Pregunta **4**

Correcta

Puntúa 1,00 sobre 1,00

Look for information about the Hibernate mapper. How many database versions does it support?

Seleccione una:

- a. Around 45. ✔
- b. The 10 most used in the market.
- c. All market databases.
- d. Only 3: HSLQDB, Oracle and MySQL.

null

La respuesta correcta es: Around 45.

Pregunta **5**

Correcta

Puntúa 1,00 sobre 1,00

Look for information: JPA mappers allow you to specify queries, but in what ways?

Seleccione una:

- a. In SQL, no more is needed; after all, there is a relational database behind.
- b. In JPQL, a specific language similar to SQL but object oriented.
- c. With a special API that allows you to build the query by creating objects that represent parts of the query.
- d. In any of the three ways cited. ✔

null

La respuesta correcta es: In any of the three ways cited.

Pregunta **6**

Correcta

Puntúa 1,00 sobre 1,00

Once an object is loaded, it is kept in memory to avoid loading it repeatedly. Likewise, in order to reduce write operations against the database the modifications are kept in memory delaying as much as possible the writing. That is, these mappers manage a cache. What kind of cache?

Seleccione una:

a. Cache per transaction, it lasts what the transaction does.

b. Cache per persistence session, it lasts what the persistence session lasts. ✔

c. Cache by Java process, lasts whatever the Java machine is started.

d. Cluster cache, there is a machine dedicated to maintaining the cache that serves several other processing machines.

null

La respuesta correcta es: Cache per persistence session, it lasts what the persistence session lasts.

Pregunta **7**

Correcta

Puntúa 1,00 sobre 1,00

Once the first object of the graph is loaded into memory, from it you can access the other associated objects. It is as if the graph was virtually loaded into memory. In what ways does the mapper get this effect?

Seleccione una:

a. Based on the principle of temporal space locality (remember operating systems), the mapper predicts which area you are going to visit next and load it along with the first object you ask for.

b. Load the object you ask for and all that are associated with it.

c. Load the object you ask and all that are associated with it and this recursively.

d. I can specify to load the objects associated with one lazily or eagerly. ✔

null

La respuesta correcta es: I can specify to load the objects associated with one lazily or eagerly.

Pregunta **8**

Correcta

Puntúa 1,00 sobre 1,00

Two concurrent users upload copies of the same entity (object). If each one modifies their version, how does the mapper relate the two copies to the row of the table from which they come?

Seleccione una:

a. It's a mapper business, it doesn't affect me at all as a programmer.

b. I must override hashCode() and equals() on the mutable attributes of the entity.

c. Because I have told the mapper what attributes of the object form the key in the database. ✔

d. It's a database thing and it doesn't affect the mapper (nor me).

null

La respuesta correcta es: Because I have told the mapper what attributes of the object form the key in the database.

Pregunta **9**

Correcta

Puntúa 1,00 sobre 1,00

Check out the Javadoc of the java.util.Set and java.util.Map interfaces. What condition objects that are stored in Set and those that are key in Map must meet?

Seleccione una:

a. It cannot be changed the value of any attribute that affects the hashCode() (and equals()) while they are contained in those collections. ✔

b. No restrictions are specified.

c. They can change the value of the hashCode() but equals() should continue to return the same result.

d. They can change the result of equals() but not that of hashCode().

null

La respuesta correcta es: It cannot be changed the value of any attribute that affects the hashCode() (and equals()) while they are contained in those collections.

Pregunta **10**

Correcta

Puntúa 1,00 sobre 1,00

Over what attributes should the methods hashCode() and equals() be defined for an object that represents an entity?

Seleccione una:

- a. It is not necessary. The default implementation is enough.
- b. Above all.
- c. About those which define its identity. ✔
- d. It is irrelevant.

null

La respuesta correcta es: About those which define its identity.

Pregunta **11**

Correcta

Puntúa 1,00 sobre 1,00

Which of the following are structural differences?

Seleccione una:

- a. Granularity, transaction control, subtype and identity.
- b. Granularity, subtype, identity and navigation.
- c. Subtype, associations, identity and granularity. ✔
- d. Concurrence, navigation and caching.

null

La respuesta correcta es: Subtype, associations, identity and granularity.

Pregunta **12**

Correcta

Puntúa 1,00 sobre 1,00

The difference by navigation refers to:

Seleccione una:

- a. Departing from an entity you can access those that are linked to it. That is possible in a graph of objects by the references between them and in the relational model by making queries with join. ✔
- b. In databases you can navigate freely using the foreign keys while in a graph of objects this is not possible since they do not exist.
- c. When the and object loaded in memory, for each association that is navigated (through getters for example) we must be explicitly indicate to the mapper at runtime to load the next objects with a call to mapper.loadNext().
- d. That difference is solved by putting cross references between objects.

null

La respuesta correcta es: Departing from an entity you can access those that are linked to it. That is possible in a graph of objects by the references between them and in the relational model by making queries with join.

Pregunta **13**

Correcta

Puntúa 1,00 sobre 1,00

If there are two concurrent users and both load common areas of the graph, how many copies of the common objects are in memory at the same time with a mapper that follows the JPA specification?

Seleccione una:

- a. There is one copy for each opened persistence session, that is, for each concurrent user. ✔
- b. Common objects are instantiated only once, that is why the methods hashCode() and equals() must be defined over the identity.
- c. It depends on the policy of navigation chosen: eager or lazy. With an eager policy, the common objects are only loaded once, with the lazy one there will be individual copies.
- d. The specification lefts this to the decision of the implementers of the mapper.

null

La respuesta correcta es: There is one copy for each opened persistence session, that is, for each concurrent user.

Pregunta **14**

Correcta

Puntúa 1,00 sobre 1,00

If there are two concurrent users, both load common areas of the graph and both modify a common entity, how does the mapper manage the collision of modifications?

Seleccione una:

- a. A transaction must be opened within each persistence session and therefore the modifications made are protected by the transaction mechanisms that we already know. ✔
- b. The user who makes the last update will crush what the other has done. The mapper has no control and that is one of the drawbacks of working with mappers. The developer has to add the control code required.
- c. It is such an unusual situation that the specification does not even bother to mention it.
- d. We avoid the situation by configuring the database to the highest transaction isolation level: SERIALIZABLE.

null

La respuesta correcta es: A transaction must be opened within each persistence session and therefore the modifications made are protected by the transaction mechanisms that we already know.

Pregunta **15**

Correcta

Puntúa 1,00 sobre 1,00

Imagine this situation with the implementation of CarWorkshop: Alice executes the use case "client management" to update one client's data because the last name is misspelled. Bob, do the same for the same client a few seconds later. Now, both Alice and Bob are seeing the (same) client's data on the screen. Then, Alice modifies the last name and when she is going to save the changes, her phone rings and she answers the call. Meanwhile, Bob modifies the last name and adds the date of birth, which was not filled in, and save the changes. When Alice finishes the call, she saves her changes, several minutes after Bob has saved his. What data is finally on the system?

Seleccione una:

- a. Hers ✔
- b. His
- c. Both combined
- d. None of them

null

La respuesta correcta es: Hers

Pregunta **16**

Correcta

Puntúa 1,00 sobre 1,00

On the previous situation, if we analyze step by step, we have the following: Alice causes the client data to be displayed on the screen, for this the system does a reading transaction (that lasts what it takes the system to recover that data (milliseconds)). A few seconds later Bob does the same. At this moment, both are seeing the same data on their screens. When Bob saves, a write transaction is executed (milliseconds). Several minutes later, Alice saves her data in another write transaction, which does not collide with any other. Which statement is true?

Seleccione una:

- a. There are two types of transactions happening at the same time: system transactions and user transactions. The former are those perceived and managed by the database. The latter are those perceived by the user (sometimes called application transactions or business transactions). For the user, editing a client is a single operation, regardless of its duration.
- b. It is usually unfeasible to match a user transaction with a system transaction. This is because the unpredictable duration of the user transaction could leave the system locked on the client he/she is editing (imagine Alice having a half hour long phone conversation, or the user leaving its place to have a coffee, or not coming back until the next day ...).
- c. System transactions (e.g. database) are only able to avoid conflicts when the transactions coincide at the same time (milliseconds time frame). Thas is not the case here.
- d. It should be the application which has to manage the collision of the user transactions.
- e. All are true. ✔

null

La respuesta correcta es: All are true.

Pregunta **17**

Correcta

Puntúa 1,00 sobre 1,00

Still with the previous situation, what solution do you think is more appropriate?

Seleccione una:

a. Do nothing. There is not going to be that coinciding situation, there is only one foreman in the workshop. And if it would be the case of having two foremen it wouldn't be such a bad thing if one has to repeat the operation (if they realize, of course...).

b. Make the application maintain a list of locked objects. If an object is on that list, it means it is being used by someone and cannot be processed. Thus, the reading operation implied by Bob would not be successful since the client would already be locked. When Alice saved the changes, the object leaves the list and Bob can proceed. To avoid the "leave for a coffee" problem that list should be scanned periodically to remove those that objects remains too long on it (a timeout). This is a pessimistic control.

c. Have the application add a version field to each entity (e.g. a counter). Thus, when Alice and Bob read the client object they both receive the (same) data with the same version. When Bob saves his changes, the application updates it and increases the version. Later Alice data arrive, but from the previous version (the version must always come and go with the data, the user cannot modify it, it is a hidden field). At this time, the application compares the version of the data that Alice sends with the most current version. If they do not match, the application aborts the transaction. Alice will have to start over. This is an optimistic control.

d. Any of the solutions may be valid. It will depend on the context of use, it is necessary to analyze it carefully. If it is true that the probability of a collision is non-existent, and if there were any, its effects did not have worrisome consequences, "Doing nothing" may be the option (not perfect but enough). There are many applications out there with this potential problem. If the effects of the lost update worry us, or we want to do things properly, we should consider one of the other two. The key will be in the frequency of the collisions and the cost of (re)doing it for the user. With high frequency and cost, it is clear that the pessimistic control is adequate. With low frequency or cost, the optimist option is the right one. In intermediate situations, you must do a calm assessment. ✔

null

La respuesta correcta es: Any of the solutions may be valid. It will depend on the context of use, it is necessary to analyze it carefully. If it is true that the probability of a collision is non-existent, and if there were any, its effects did not have worrisome consequences, "Doing nothing" may be the option (not perfect but enough). There are many applications out there with this potential problem. If the effects of the lost update worry us, or we want to do things properly, we should consider one of the other two. The key will be in the frequency of the collisions and the cost of (re)doing it for the user. With high frequency and cost, it is clear that the pessimistic control is adequate. With low frequency or cost, the optimist option is the right one. In intermediate situations, you must do a calm assessment.

◄ Algunas preguntas interesantes

Ir a...

Questionnaire. Persistent object management. ►