

LO54 Examination

Documents autorisés / Authorized documents : Aucun / None

Durée / Duration : 1h15.

Notation : 1 question = 1 point

Des mots clés sont attendus, l'exhaustivité des mots clés et la clarté de la réponse sont nécessaires à l'obtention du point complet.

Key words are expected, the completeness of the key words and the clarity of the answer are necessary to obtain the complete point.

Question 1 : A quoi sert Maven ?

What the goal of maven?

Question 2 : Qu'est-ce qu'un repository Maven ?

What is a Maven repository?

Question 3 : Que désigne la balise <packaging> dans le pom.xml ?

What is the <packaging> tag for in a pom.xml file?

Question 4 : Ou maven va-t-il chercher ses artifacts ?

Where maven search his artifact.

Question 5 : Que fait la commande "mvn install" ?

What do 'mvn install' ?

Question 6 : Expliquez ce qu'est SOA.

Describe what SOA is ?

Question 7 : Expliquez ce qu'est un driver JDBC

Describe what driver JDBC is ?

Question 8 : Expliquez les limites de l'usage de JDBC

Explain limit of JDBC?

Question 9 : Quels sont les avantages de l'utilisation d'un PreparedStatement ?

What advantages of a PreparedStatement ?

Question 10 : Où place-t-on les fichiers de configuration d'Hibernate

Where are hibernate configuration files ?

Question 11 : Expliquez ce qu'est le dialect Hibernate

Describe what Hibernate dialect is ?

Question 12 : À quoi sert la propriété hbm2ddl.auto' de Hibernate?

What the aim of the Hibernate 'hbm2ddl.auto' property?

Question 13 : Expliquez ce qu'est le Lazy Loading

Describe what Lazy Loading is ?

Question 14 : Qu'est-ce que le HQL ?

Describe what HQL is ?

Question 15: À quoi sert le caractère '?' dans un requête HQL ?

Describe what '?' is in a HQL query ?

Question 16 : Qu'est-ce qu'un ORM ?

What is an ORM ?

Question 17 : Quels sont les avantages à utiliser un ORM ?

What advantages to use an ORM ?

Question 18 : Quelle requête SQL est exécutée par Hibernate avec le code qui suit ?

What SQL query is going to be executed with the following Hibernate snippet ?

```
public void function(String value) {
    Session session = HibernateUtil.getSessionFactory().openSession();
    try {
        Query q = session.createQuery("from Person where id=5");
        Person p = (Person) query.uniqueResult();
        p.setId(5);
        p.setName(value);
    }...
```

Question 19 : Quelles requêtes SQL seront exécutées par Hibernate avec le code qui suit ?

(Le champ id de la class Person est une colonne auto-incrémentée)

Quel est le résultat dans la base de donnée ?

What SQL queries are going to be executed with the following Hibernate snippet ?

(The filed id of Person class is auto_increment)

What is the result in database ?

```
public void function(String value) {
    Session session = HibernateUtil.getSessionFactory().openSession();
    Person p = new Person();
    p.setName(value);
    try {
        session.beginTransaction();
        session.persist(p);
        session.getTransaction().commit();
    }...
```

Question 20 : Décrivez le DDL (requête de création ou modèle physique de données) des tables mappées dans l'exemple ci-dessous (Nom des tables, nom et lorsque cela est possible le type des colonnes). Côté Java, les 2 entités en question n'ont d'autre lien que de l'une vers l'autre.

Describe the DDL (creation request or MPD) of the tables mapped through the following mapping files

(Tables name, columns' names and when possible data type). Regarding the Java code, the 2 entities only refers each other (no extra relationships).

```
<hibernate-mapping package="fr.utbm.entity">
  <class name="User">
    <id name="identifiant" column="id">
      <generator class="identity"/>
    </id>
    <property name="username" not-null="true" />
    <property name="Valid" not-null="true" />
    <property name="email" not-null="true" />
    <many-to-one name="currentProject" column="currentproject_id" unique="true"/>
  </class>
  <class name="Project" table="project">
    <id name="id" column="project_id">
      <generator class="assigned"/>
    </id>
    <property name="projectcode" not-null="true" />
    <property name="projectlabel" not-null="true" />
    <join table="projects">
      <key column="project_id" unique="true"/>
      <many-to-one name="user" column="identifiant" not-null="true"/>
    </join>
  </class>
</hibernate-mapping>
```

Question 20 (Bonus) : Écrire les class Java User et Project

Write java classes User and Project