|  |  |  |
| --- | --- | --- |
| **Q 1** | What is ORM?   1. Object Role Modeling. 2. Object Rational Model 3. Operational Resources Modeling 4. Object Relational Mapping | S |
| **Q2.** | What will be the result of the following code?  Configuration config = **new** Configuration();  config.configure(“hibernate.cfg.xml”);    /\* Build session factory \*/  sessions = config.buildSessionFactory();  Where should hibernate.cfg.xml file ideally be located?   1. At the root of the application 2. Besides entity classes 3. At classpath 4. Anywhere in application. Hibernate finds it out. | S |
| **Q 3** | Which one out of following is an ORM tool?   1. JDBC c) Hibernate 2. Spring JDBC d) Java IO. | S |
| Q4. | Which state of object ensures synchronization with DB when object contents are changed?   1. Persistent c) Serializable 2. Transient d) Detached. | M |
| Q5. | Which method out of following detaches Persistent object from session.   1. The saveAndUpdate() c) The detach() 2. The delete() d) The lock() | C |
| **Q 6** | Why following object is not persisted to the table successfully?  Emp emp = new Emp();  emp.setId(123);  emp.setName(“Chandra”);  tx.commit();   1. The object is transient c) The object is detached 2. The object is persistent d) The object is not serializable | M |
| Q 7 | Which method out of following converts object from Transient to Persistent state?   1. The delete() c) The save() 2. The merge() d) The createQuery() | M |
| Q 8 | Find two options for- when are the changed contents of persistent object written to the database?   1. At the time of opening session 2. At the time of closing session 3. At the time of flushing session 4. At the time of committing session. | M |
| Q 9 | Which ONE is the most suitable answer for NOT being the correct difference between SessionFactory and Session.   1. SessionFactory is heavy weight while Session is light weight 2. SessionFactory has a factory method to create Session 3. SessionFactory is thread safe while session is not 4. None of these. | S |
| Q 10 | How can a field be prevented for updation in Hibernate?   1. Set *update=’false’* in class tag. 2. Set *update=’false’* in property tag. 3. Set *dynamic-insert=’false’* in class tag. 4. Set *dynamic-update=’false’* in class tag. | C |
| Q 11 | The weakness of ‘Table for each concrete class’ is…   1. Does not repeat common columns in each table. 2. Needs an additional Discriminator column 3. Does not support simple association 4. Does not support polymorphic association. | M |
| Q 12 | Which database feature can’t be mapped to an entity or its property in Hibernate?   1. Composite Key c) Triggers 2. Many-Many association d) Formula | C |
| Q 13 | 1. Select the odd man out with respect to number of tables needed at DB side? 2. One-One c) Many-One 3. One-Many d) Many-Many | S |
| Q14 | 1. Which XML attribute is responsible to save/update/delete parent-child graph in database? 2. inverse c) cascade 3. length d) fetch | C |
| Q15 | What is Criteria query?   1. The Object oriented query not designed dynamically. 2. The Object oriented query designed dynamically. 3. The database specific query. 4. Query named and stored in xml mapping file. | C |
| Q 16 | Which is the default fetch mode of loading collection elements in Hibernate?   1. The Eager c) The Lazy 2. The Immediate d) None of these | S |
| Q 17 | Which sentence closely defines Bags?   1. The un-ordered collection of unique elements. 2. The ordered collection allows duplicates 3. The ordered collection of unique elements 4. The key-value map | S |
| Q 18 | Which collection needs index to sequence entities?   1. The ‘set’ c) The ‘list’ 2. The ‘bag’ d) The ‘map’ | S |
| Q 19 | Observe following criteria. Find the nearest option out of following…  Criterion restriction = Expression.between("empNo", 7800, 7900);  List<Emp> empList = session.createCriteria(Emp.class)  .add(restriction)  .list();   1. Query for all records with empNo in 7800 or 7900 2. Query for all records with empNo including 7800, 7900. 3. Query for all records with empNo between 7800 to 7900. 4. Query for all records excluding empNo- 7800, 7900. | M |
| Q 20 | Observe following query. Find the nearest option out of following…  String qry = “from Emp where eName like :name And job like :job"”;  Query qryForAllEmployees = session.createQuery(qry);  qryForAllEmployees.setString("name", "J%");  qryForAllEmployees.setString("job", "MANAGER");  List<Emp> empList = qryForAllEmployees.list();   1. Query for all records for name like ‘name’ and job like ‘job’. 2. Query for all Managers with name starts with ‘J’. 3. Query for all records including name and job as null. 4. Query for all records. | M |
| Q 21 | Which type of data is most suitable to cache in IInd level Cache with 'Read Only' strategy?   1. Objects for which occasional dirty read is allowed. 2. Data for which frequent read-write allowed. 3. Data for which transactional read-write is allowed. 4. Data which is rarely changed. | C |
| Q 22 | Which TWO of the following are applicable reasons for an object existing in Cache Level-II but not in Cache Level I?   1. Object is evicted from Level I. 2. Least recently used object automatically dropped from Level I to make space available to other objects. 3. New session started which yet to get object from Level-II 4. It is not possible for an object to be in Level-II without in Level-I. | C |
| Q 23 | The reason for preferring Version numbering over Timestamp versioning is...   1. Version number is simpler compared to Timestamp versioning. 2. Perfection of Timestamp versioning depends on time granularity supported by Data Base while Version numbering is robust. 3. Version number offers better scalability compared to Timestamp version. 4. Version number allows pessimistic locking also. | C |
| Q 24 | Which statement out of following is nearest to explain following configuration?  <property name=*"hibernate.jdbc.batch\_size"*>20</property>   1. Defines batch size and automatically flushes batch queries when batch is full. 2. Defines batch size and automatically flushes batch queries when session is closed. 3. Defines JDBC buffer of given size for a batch. Flushing to be done explicitly. 4. None of these. | M |
| Q 25 | 1. Hibernate support for offering services to SOAP layer? 2. Hibernates data centric approach. 3. Designing interceptors to do necessary conversion. 4. Hibernates multi-tenancy support 5. Hibernates dynamic support for DOM4J. | C |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Que** | **Ans** | **Que** | **Ans** | **Que** | **Ans** | **Que** | **Ans** | **Que** | **Ans** |
| 1 | d | 6 | a | 11 | d | 16 | c | 21 | d |
| 2 | c | 7 | c | 12 | c | 17 | a | 22 | a, c |
| 3 | c | 8 | c, d | 13 | d | 18 | c | 23 | b |
| 4 | a | 9 | d | 14 | c | 19 | c | 24 | c |
| 5 | b | 10 | b | 15 | b | 20 | b | 25 | d |