Ashu Jauhari

[Email address]

Hibernate Question Bank

Table of Contents

[JPA Hibernate MCQ -Set -1 3](#_Toc151733989)

[1. What is Hibernate? 3](#_Toc151733990)

[2. Which of the following is true about Hibernate? 3](#_Toc151733991)

[3. What is the purpose of Hibernate's SessionFactory? 3](#_Toc151733992)

[4. Which configuration file is used to configure Hibernate settings? 3](#_Toc151733993)

[5. What is the purpose of Hibernate's Session object? 3](#_Toc151733994)

[6. What is the default primary key generation strategy provided by Hibernate? 3](#_Toc151733995)

[7. Which annotation is used to specify the primary key in a Hibernate entity class? 4](#_Toc151733996)

[8. What is the purpose of Hibernate's HQL (Hibernate Query Language)? 4](#_Toc151733997)

[9. What is lazy loading in Hibernate? 4](#_Toc151733998)

[10. Which Hibernate mapping type is used to represent a one-to-many association? 4](#_Toc151733999)

[11. What is the purpose of Hibernate's EntityManager? 4](#_Toc151734000)

[12. Which database operations does Hibernate's EntityManager support? 5](#_Toc151734001)

[13. Which annotation is used to map a Java class to a database table in Hibernate? 5](#_Toc151734002)

[14. What is the purpose of Hibernate's Second Level Cache? 5](#_Toc151734003)

[15. What is the role of Hibernate's Transaction object? 5](#_Toc151734004)

[16. Which association mapping type is used to represent a many-to-many association in Hibernate? 5](#_Toc151734005)

[17. What is the purpose of Hibernate's Criteria API? 5](#_Toc151734006)

[18. Which annotation is used to specify the column name in a Hibernate entity class? 6](#_Toc151734007)

[19. What is the purpose of Hibernate's Dialect class? 6](#_Toc151734008)

[20. Which configuration method is used to obtain a Hibernate SessionFactory object? 6](#_Toc151734009)

[21. Which Hibernate annotation is used to enable caching for an entity class? 6](#_Toc151734010)

[22. What is the difference between EAGER and LAZY fetching in Hibernate? 6](#_Toc151734011)

[23. What are Hibernate Interceptors used for? 7](#_Toc151734012)

[Answers and Explanations 7](#_Toc151734013)

[Question 1 7](#_Toc151734014)

[Question 2 7](#_Toc151734015)

[Question 3 7](#_Toc151734016)

[Question 4 7](#_Toc151734017)

[Question 5 8](#_Toc151734018)

[Question 6 8](#_Toc151734019)

[Question 7 8](#_Toc151734020)

[Question 8 8](#_Toc151734021)

[Question 9 8](#_Toc151734022)

[Question 10 8](#_Toc151734023)

[Question 11 9](#_Toc151734024)

[Question 12 9](#_Toc151734025)

[Question 13 9](#_Toc151734026)

[Question 14 9](#_Toc151734027)

[Question 15 9](#_Toc151734028)

[Question 16 9](#_Toc151734029)

[Question 17 10](#_Toc151734030)

[Question 18 10](#_Toc151734031)

[Question 19 10](#_Toc151734032)

[Question 20 10](#_Toc151734033)

[Question 21 10](#_Toc151734034)

[Question 22 10](#_Toc151734035)

[Question 23 11](#_Toc151734036)

[**Answer : D** 12](#_Toc151734037)

[**Answer : A** 12](#_Toc151734038)

[**Answer : A** 12](#_Toc151734039)

[**Answer : C** 12](#_Toc151734040)

[**Answer : A** 13](#_Toc151734041)

[**Answer : A** 13](#_Toc151734042)

[**Answer : D** 13](#_Toc151734043)

[**Answer : A** 13](#_Toc151734044)

[Interview Questions 24](#_Toc151734045)

[**1. What do you mean by the Entity in JPA?** 24](#_Toc151734046)

[**2. What is meant by the attribute of the entity class in the terminology of JPA?** 24](#_Toc151734047)

[**3. List some JPA requirements for entity classes.** 24](#_Toc151734048)

[**4. What are the types of data that the entity class attributes allow?** 24](#_Toc151734049)

[**5. What are the embeddable classes?** 24](#_Toc151734050)

[**6. List the valid attribute types that are included in the primary key of the entity class so that you can use them for any database.** 25](#_Toc151734051)

[**7. What are the requirements set by JPA for embeddable classes?** 25](#_Toc151734052)

[**8. List the four lifecycle statuses of the life cycle of an entity instance.** 26](#_Toc151734053)

[**9. What is an Entity Manager?** 26](#_Toc151734054)

[**10. What are the main functions of Entity Manager?** 26](#_Toc151734055)

[**11. What is meant by Mapped Superclass?** 26](#_Toc151734056)

[**12. What three inheritance mapping strategies does the JPA describe?** 27](#_Toc151734057)

[**13. List the types of connections available between the entities.** 28](#_Toc151734058)

[**14. Which two kinds of fetch strategies are available in JPA?** 28](#_Toc151734059)

[**15. Can some other embeddable class be contained in the embeddable class?** 28](#_Toc151734060)

# JPA Hibernate MCQ -Set -1

# 1. What is Hibernate?

a) A programming language

b) A database management system

c) An object-relational mapping (ORM) framework

d) A scripting language

# 2. Which of the following is true about Hibernate?

a) It is specific to a particular database management system

b) It provides a way to define database schemas

c) It simplifies object-oriented programming in Java

d) It is primarily used for front-end web development

# 3. What is the purpose of Hibernate's SessionFactory?

a) To create and manage database connections

b) To define database schemas

c) To provide a mapping between Java classes and database tables

d) To handle database transactions

# 4. Which configuration file is used to configure Hibernate settings?

a) hibernate.cfg.xml

b) web.xml

c) persistence.xml

d) application.properties

# 5. What is the purpose of Hibernate's Session object?

a) To execute SQL queries

b) To perform database CRUD operations

c) To manage database transactions

d) To define Hibernate entity mappings

# 6. What is the default primary key generation strategy provided by Hibernate?

a) AUTO

b) IDENTITY

c) SEQUENCE

d) TABLE

# 7. Which annotation is used to specify the primary key in a Hibernate entity class?

a) @PrimaryKey

b) @Id

c) @GeneratedValue

d) @Key

# 8. What is the purpose of Hibernate's HQL (Hibernate Query Language)?

a) To define database schemas

b) To create Java objects from database tables

c) To perform database CRUD operations

d) It allows developers to write database queries using entity and property names

# 9. What is lazy loading in Hibernate?

a) Loading data from the database only when needed

b) Loading all data eagerly during initialization

c) Loading data in a separate thread

d) Loading data from a cache

# 10. Which Hibernate mapping type is used to represent a one-to-many association?

a) @OneToMany

b) @ManyToOne

c) @OneToOne

d) @ManyToMany

# 11. What is the purpose of Hibernate's EntityManager?

a) To manage database connections

b) To define database schemas

c) To handle database transactions

d) To persist and retrieve entities

# 12. Which database operations does Hibernate's EntityManager support?

a) Create and update

b) Read and delete

c) Create, read, update, and delete

d) Read and update

# 13. Which annotation is used to map a Java class to a database table in Hibernate?

a) @Entity

b) @Table

c) @Column

d) @Id

# 14. What is the purpose of Hibernate's Second Level Cache?

a) To cache database query results

b) To cache Java objects

c) To cache database connections

d) To cache database schema information

# 15. What is the role of Hibernate's Transaction object?

a) To execute SQL queries

b) To manage database connections

c) To handle database transactions

d) To define Hibernate entity mappings

# 16. Which association mapping type is used to represent a many-to-many association in Hibernate?

a) @OneToMany

b) @ManyToOne

c) @OneToOne

d) @ManyToMany

# 17. What is the purpose of Hibernate's Criteria API?

a) To define database schemas

b) To perform database CRUD operations

c) To execute SQL queries

d) To build type-safe queries using Java code

# 18. Which annotation is used to specify the column name in a Hibernate entity class?

a) @PrimaryKey

b) @Id

c) @GeneratedValue

d) @Column

# 19. What is the purpose of Hibernate's Dialect class?

a) To define database schemas

b) To create database connections

c) To handle database transactions

d) To provide database-specific SQL dialects

# 20. Which configuration method is used to obtain a Hibernate SessionFactory object?

a) configure()

b) buildSessionFactory()

c) createSessionFactory()

d) getSessionFactory()

# 21. Which Hibernate annotation is used to enable caching for an entity class?

a) @Cacheable

b) @Cached

c) @Cache

d) @Caching

# 22. What is the difference between EAGER and LAZY fetching in Hibernate?

a) EAGER fetch loads associated entities eagerly, while LAZY fetch loads them lazily on demand

b) EAGER fetch loads associated entities on demand, while LAZY fetch loads them eagerly

c) EAGER fetch loads only the primary entity, while LAZY fetch loads the associated entities as well

d) EAGER fetch and LAZY fetch have no difference in Hibernate

# 23. What are Hibernate Interceptors used for?

a) To intercept and modify SQL queries before execution

b) To intercept and modify entity properties before persistence

c) To intercept and modify database transactions

d) To intercept and modify Hibernate session operations

# Answers and Explanations

## Question 1

**Answer:**

c) An object-relational mapping (ORM) framework.

**Explanation:**

Hibernate is an ORM framework that provides a way to map Java objects to relational database tables, simplifying database interaction in Java applications.

## Question 2

**Answer:**

c)It simplifies object-oriented programming in Java.

**Explanation:**

Hibernate simplifies database access by abstracting away the complexities of JDBC (Java Database Connectivity) and providing an object-oriented approach to interacting with databases.

## Question 3

**Answer:**

a) To create and manage database connections.

**Explanation:**

The **[SessionFactory](https://www.javaguides.net/2018/11/hibernate-hello-world-tutorial.html" \t "_blank)** in Hibernate is responsible for creating and managing database connections. It is used to configure Hibernate settings and create sessions for performing database operations.

## Question 4

**Answer:**

a) hibernate.cfg.xml.

**Explanation:**

The *hibernate.cfg.xml* configuration file is used to configure Hibernate settings, such as database connection details, dialect, mapping files, and other properties required for Hibernate to work correctly.

## Question 5

**Answer:**

c) To manage database transactions.

**Explanation:**

The Session object in Hibernate represents a single unit of work and is responsible for managing database transactions. It provides methods for performing CRUD (Create, Read, Update, Delete) operations on entities.

## Question 6

**Answer:**

a) AUTO.

**Explanation:**

The AUTO primary key generation strategy allows Hibernate to choose the appropriate strategy based on the underlying database. It typically uses identity columns, sequences, or table-based strategies for generating primary keys.

## Question 7

**Answer:**

b) @Id.

**Explanation:**

The *@Id* annotation is used to specify the primary key in a Hibernate entity class. It marks a field or a property as the identifier of the entity.

## Question 8

**Answer:**

d) To execute SQL queries.

**Explanation:**

[**Hibernate Query Language (HQL)**](https://www.javaguides.net/2019/10/hibernate-query-language-tutorial.html) is a powerful object-oriented query language similar to SQL. It allows developers to write database queries using entity and property names instead of database tables and column names.

## Question 9

**Answer:**

a) Loading data from the database only when needed.

**Explanation:**

Lazy loading is a technique in Hibernate where associated data is fetched from the database only when it is accessed for the first time. This improves performance by reducing unnecessary database queries.

## Question 10

**Answer:**

a) @OneToMany.

**Explanation:**

The [**@OneToMany**](https://www.javaguides.net/2019/08/jpa-hibernate-one-to-many-unidirectional-mapping-example.html) annotation is used to represent a one-to-many association in Hibernate. It is used to define a relationship where one entity has a collection of other entities.

## Question 11

**Answer:**

d) To persist and retrieve entities.

**Explanation:**

The EntityManager in Hibernate is responsible for persisting and retrieving entities from the database. It provides methods for managing the lifecycle of entities and performing CRUD operations.

## Question 12

**Answer:**

c) Create, read, update, and delete.

**Explanation:**

Hibernate's EntityManager supports all the major database operations, including creating new entities, reading existing entities, updating entities, and deleting entities.

## Question 13

**Answer:**

a) @Entity.

**Explanation:**

The *@Entity* annotation is used to map a Java class to a database table in Hibernate. It marks a class as an entity that can be persisted in the database.

## Question 14

**Answer:**

b) To cache Java objects.

**Explanation:**

Hibernate's Second Level Cache is used to cache Java objects to improve performance. It reduces the need to fetch the same objects from the database repeatedly.

## Question 15

**Answer:**

c) To handle database transactions.

**Explanation:**

The Transaction object in Hibernate is responsible for managing database transactions. It provides methods for beginning, committing, or rolling back transactions.

## Question 16

**Answer:**

d) @ManyToMany.

**Explanation:**

The [**@ManyToMany**](https://www.javaguides.net/2019/12/hibernate-5-many-to-many-annotation-mapping-example.html) annotation is used to represent a many-to-many association in Hibernate. It is used to define a relationship where multiple entities can be associated with multiple other entities.

## Question 17

**Answer:**

d) To build type-safe queries using Java code.

**Explanation:**

Hibernate's Criteria API provides a programmatic way to create database queries using Java code. It allows developers to build type-safe queries without writing SQL directly.

## Question 18

**Answer:**

d) @Column.

**Explanation:**

The *@Column* annotation is used to specify the column name in a Hibernate entity class. It allows developers to map a field or property to a specific database column.

## Question 19

**Answer:**

d) To provide database-specific SQL dialects.

**Explanation:**

Hibernate's Dialect class provides database-specific SQL dialects. It generates the appropriate SQL statements and handles the differences in SQL syntax and behavior among various database systems.

## Question 20

**Answer:**

b) buildSessionFactory().

**Explanation:**

The *buildSessionFactory()* method is used to obtain a Hibernate SessionFactory object. It is typically called during application initialization to create a SessionFactory instance that can be used throughout the application.

## Question 21

**Answer:**

a) @Cacheable.

**Explanation:**

The *@Cacheable* annotation is used to enable caching for an entity class. It indicates that the entity should be cached, improving performance by reducing database access.

## Question 22

**Answer:**

a) EAGER fetch loads associated entities eagerly, while LAZY fetch loads them lazily on demand.

**Explanation:**

EAGER fetching in Hibernate loads associated entities immediately along with the primary entity, while LAZY fetching loads them only when they are accessed for the first time.

## Question 23

Answer:

b) To intercept and modify entity properties before persistence.

# JPA Hibernate MCQ -Set -2

Q 1 - Which of the following is true about Hibernate?

[A - Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code.](javascript:void(0);)

[B - Hibernate provides simple APIs for storing and retrieving Java objects directly to and from the database.](javascript:void(0);)

[C - Hibernate abstracts away the unfamiliar SQL types and provide us to work around familiar Java Objects.](javascript:void(0);)

[D - All of the above.](javascript:void(0);)

### **Answer : D**

Q 2 - Which of the following object is used to create SessionFactory object in hibernate?

[A - Configuration](javascript:void(0);)

[B - Session](javascript:void(0);)

[C - SessionFactory](javascript:void(0);)

[D - Transaction](javascript:void(0);)

### **Answer : A**

Q 3 - Which of the following is true about hibernate.dialect property in hibernate configuration?

[A - This property makes Hibernate generate the appropriate SQL for the chosen database.](javascript:void(0);)

[B - This property makes Hibernate generate the appropriate java code for the chosen database.](javascript:void(0);)

[C - Both of the above.](javascript:void(0);)

[D - None of the above](javascript:void(0);)

### **Answer : A**

Q 4 - Which method is used to update the state of the given instance from the underlying database?

[A - Session.store()](javascript:void(0);)

[B - Session.keep()](javascript:void(0);)

[C - Session.update()](javascript:void(0);)

[D - Session.load()](javascript:void(0);)

### **Answer : C**

Q 5 - Which of the following element is used to represent many-to-one relationship in hibernate?

[A - <many-to-one>](javascript:void(0);)

[B - <many-one>](javascript:void(0);)

[C - <ManyToOne>](javascript:void(0);)

[D - None of the above](javascript:void(0);)

### **Answer : A**

Q 6 - What is value of lazy loading by default?

[A - true](javascript:void(0);)

[B - false](javascript:void(0);)

### **Answer : A**

Q 7 - Which of the following is true about HQL?

[A - HQL takes java objects in the same way as SQL takes tables.](javascript:void(0);)

[B - HQL is a Object Oriented Query language](javascript:void(0);)

[C - HQL is database independent.](javascript:void(0);)

[D - All of the above.](javascript:void(0);)

### **Answer : D**

Q 8 - Which of the following is true about hibernate annotations?

[A - Hibernate Annotations is the powerful way to provide the metadata for the Object and Relational Table mapping.](javascript:void(0);)

[B - All the metadata is clubbed into the POJO java file along with the code this helps the user to understand the table structure and POJO simultaneously during the development.](javascript:void(0);)

[C - Both of the above.](javascript:void(0);)

[D - None of the above.](javascript:void(0);)

### **Answer : A**

# JPA Hibernate MCQ -Set -3

Q1 What does ORM stand for?

**(a) Object Relational Model**

**(b) Object Resource Mapping**

**(c) Object Relational Mapping**

**(d) Object Reference Mapping**

Answer: c) Object Relational Mapping

**Q#2.** Which of the following technologies is not an ORM framework?

**(a) Hibernate**

**(b) TopLink**

**(c) MyBatis**

**(d) Gradle**

Answer: d) Gradle

**Q#3.** What is the purpose of Hibernate in Java applications?

**(a) To achieve database connections**

**(b) To simplify database querying using ORM technique**

**(c) To manage user authentication**

**(d) To increase application performance**

Answer: b) To simplify database querying using ORM technique

**Q#4.** Bob is working in a hibernate based project. He has two entities Employee and Address with the relation Employee  ‘HAS-A’  Address. He has a requirement where in he wants to delete address as well on deletion of Employee. Which of the following Cascade Type should he use?

**(a) CascadeType.REMOVE**

**(b) CascadeType.MERGE**

**(c) CascadeType.DELETE**

**(d) CascadeType.REFRESH**

Answer: c) CascadeType.DELETE

**Q#5.** Which of the following is a feature of Hibernate?

**(a) Automatic database schema generation**

**(b) Automatic memory management**

**(c) Dynamic class loading**

**(d) Garbage collection**

Answer: a) Automatic database schema generation

**Q#6.** What does JPA stand for?

**(a) Java Persistence API**

**(b) Java Programming Architecture**

**(c) Java Persistence Algorithm**

**(d) Java Persistence Architecture**

Answer: a) Java Persistence API

**Q#7.** Which of the following is not a key component of JPA?

**(a) Entity Manager**

**(b) Persistence Unit**

**(c) Query Language**

**(d) Session Factory**

Answer: d) Session Factory

**Q#8.** What is the main role of the EntityManager in JPA?

**(a) To achieve the database connections**

**(b) To handle entity persistence and retrieval**

**(c) To offer SQL query execution**

**(d) To manage the user session**

Answer: b) To handle entity persistence and retrieval

**Q#9.** Which of the following is incorrect about @Table annotation in JPA?

**(a) It is used to create a user defined table in JPA.**

**(b) It is used in conjunction with @Entity annotation.**

**(c) It creates user defined columns against each table.**

**(d) It offers a name attribute to specify a table name.**

Answer: c) It creates user defined columns against each table.

**Q#10.** What is a persistence unit in JPA?

**(a) A set of all database tables**

**(b) A logical group of related entities**

**(c) A container-managed entity manager**

**(d) A configuration file for all database tables**

Answer: b) A logical group of related entities

**Q#11.** Which of the following is not a valid statement about JPQL (Jakarta Persistence Query Language)?

**(a) It can be used with any type of database such as MySQL, Oracle (database independent).**

**(b) It can perform join operations.**

**(c) It can't perform an aggregate function with sorting and grouping clauses.**

**(d) It can provide single and multiple value result types.**

Answer: c) It can’t perform an aggregate function with sorting and grouping clauses.

**Q#12.** What is incorrect about a many-to-many relationship?

**(a) A many-to-many relationship occurs when multiple records in a table are associated with multiple records in another table.**

**(b) A many-to-many relationship exists between customers and products if customers can purchase various products, and products can be purchased by many customers.**

**(c) Many-to-Many relationships are lazy by default.**

**(d) A many-to-many relationship can't produce the data redundancy problem.**

Answer: d) A many-to-many relationship can’t produce the data redundancy problem.

**Q#16.**Which operator will you use for named parameter binding in JPQL (Jakarta Persistence Query Language) ?

**(a) =**

**(b) :=**

**(c) ?**

**(d) :**

Answer: d) :

**Q#17.** What is the purpose of the Hibernate SessionFactory?

**(a) Performs CRUD-based operations.**

**(b) Provides methods such as save, delete and update, retrieve.**

**(c) Responsible for the creation of Session objects.**

**(d) Manages the transactions.**

Answer: c) Responsible for the creation of Session objects.

**Q#18.**Which statement is correct about Hibernate and JPA?

**(a) Hibernate is a particular implementation of JPA.**

**(b) Hibernate is used for database connection management, while JPA is used for entity mapping.**

**(c) Hibernate supports Java language features beyond the JPA.**

**(d) Hibernate and JPA are two different ORM frameworks for different programming languages.**

Answer: a) Hibernate is a particular implementation of JPA.

**Q#19.**Which of the following annotations is used to mark a field as the primary key in JPA?

**(a) @Id**

**(b) @PrimaryKey**

**(c) @Key**

**(d) @Primary**

Answer: a) @Id

**Q#20.**Which of the following type can’t be a primary key field in JPA?

**(a) java.sql.Date**

**(b) Any primitive wrapper**

**(c) java.math.BigInteger**

**(d) java.util.Collection**

Answer: d) java.util.Collection

**Q#21.**Which of the following is incorrect about FetchType in JPA?

**(a) FetchType is an enumerated type in the JPA Specification.**

**(b) It specifies whether the field or property should be lazily loaded or eagerly loaded.**

**(c) By default, @OneToMany associations use the FetchType.EAGER strategy.**

**(d) It can be specified for associations at the time of mapping the association.**

Answer: c) By default, @OneToMany associations use the FetchType.EAGER strategy.

**Q#22.**Which of the following annotations is used to specify the name of the table associated with an entity in Hibernate?

**(a) @TableName**

**(b) @Table**

**(c) @EntityTable**

**(d) @Entity**

Answer: b) @Table

**Q#23.**What is the purpose of the @GeneratedValue annotation in Hibernate?

**(a) To specify the strategy for primary key generation**

**(b) To define the mapping between entities and database tables**

**(c) To indicate the relationship between two entities**

**(d) To create the caching configuration for entities**

Answer: a) To specify the generation strategy for primary keys

**Q#24.**What is the purpose of the Hibernate dialect?

**(a) To define the mapping metadata for entities**

**(b) To provide the connection pooling mechanism**

**(c) To specify the database-specific SQL syntax**

**(d) To configure the second-level cache**

Answer: c) To specify the database-specific SQL syntax

**Q#25.**Which of the following is a disadvantage of using Hibernate?

**(a) Increase in productivity of developers**

**(b) Database Independency**

**(c) Performance overhead due to object-relational mapping**

**(d) Improved maintainability and code readability**

Answer: c) Performance overhead due to object-relational mapping

**Q#26.**How can you create a native SQL query in Hibernate?

**(a) Using the createNativeQuery() method**

**(b) Using the createSQLQuery() method**

**(c) Using the executeQuery() method**

**(d) Using the executeNativeQuery() method**

Answer: b) Using the createSQLQuery() method

**Q#27.**What is the purpose of the @EmbeddedId annotation in Hibernate?

**(a) It specifies the primary key for an entity using an embedded object.**

**(b) It indicates that the entity is Embeddable.**

**(c) It defines the relationship between two entities.**

**(d) It configures the caching strategy for an entity.**

Answer: a) It specifies the primary key for an entity using an embedded object

**Q#28.**In order to make an entity eligible for second-level caching, which annotation will you use in Hibernate?

**(a) @Cacheable**

**(b) @Cache**

**(c) @SecondLevelCache**

**(d) @CacheableEntity**

Answer: b) @Cache

**Q#29.**When should you use @Temporal annotation?

**(a) If you have fields with type 'java.util.Collection'.**

**(b) If you want to create a temporary field.**

**(c) If you have fields with type 'java.util.Date' or 'java.util.Calendar'.**

**(d) If you want a field to be used in the persistence context.**

Answer: c) If you have fields with type ‘java.util.Date’ or ‘java.util.Calendar’.

**Q#30.**How can you get an entity by its primary key in JPA?

**(a) By calling find() method of the EntityManager**

**(b) By calling get() method of the EntityManager**

**(c) By calling load() method of the EntityManager**

**(d) By calling key() method of the EntityManager**

Answer: a) By calling find() method of the EntityManager

**Q#31.**What is the purpose of the CascadeType.PERSIST option in JPA?

**(a) It defines that entity deletion should be cascaded to related entities.**

**(b) It defines that entity updates should be cascaded to related entities.**

**(c) It defines that entity insertion should be cascaded to related entities.**

**(d) All of the above**

Answer: c) It defines that entity insertion should be cascaded to related entities.

**Q#32.**What is the main usage of the @JoinColumn annotation in JPA?

**(a) To specify the database table and column name**

**(b) To specify a column for joining an entity association or element collection**

**(c) To define the primary key of a table in the database**

**(d) To configure the database join column**

Answer: b) To specify a column for joining an entity association or element collection

**Q#33.**What does the following property mean?

**<property name=”hbm2ddl.auto”>create</property>**

**(a) Update the database schema without re-creating it on startup**

**(b) Drop and re-create the database schema on startup**

**(c) Update the database schema on startup**

**(d) All of the above**

Answer: b) Drop and re-create the database schema on startup

**Q#34.**Which one is incorrect about Criteria Queries?

**(a) Criteria queries are more flexible as compared to HQL and JPQL.**

**(b) Like HQL and JPQL, Criteria queries provide native query support.**

**(c) Criteria queries provide better support for writing dynamic queries.**

**(d) It is easier to detect errors in the Criteria API during the compile time.**

Answer: b) Like HQL and JPQL, [Criteria queries](https://docs.jboss.org/hibernate/orm/3.2/api/org/hibernate/Criteria.html) provide native query support.

**Q#35.**Which of the following is incorrect about the Hibernate HQL (Hibernate Query Language)?

**(a) HQL queries are translated by Hibernate into traditional SQL queries.**

**(b) It supports object-oriented queries.**

**(c) It works with persistent objects and their properties.**

**(d) It works on tables and columns.**

Answer: d) It works on tables and columns

# JPA Hibernate MCQ -Set -4

1. What is the purpose of the CascadeType.ALL in JPA?  
   a. It specifies the type of the relationship between entities.  
   b. It specifies the fetch type of the relationship.  
   c. It specifies the actions to be cascaded from parent to child entities.  
   d. It specifies the order in which entities are loaded.  
   Answer: c. It specifies the actions to be cascaded from parent to child entities.

Explanation: The CascadeType.ALL in JPA specifies the actions to be cascaded from parent to child entities. For example, if a parent entity is deleted, all related child entities will also be deleted.

1. Which of the following is not a valid inheritance strategy in JPA?  
   a. SINGLE\_TABLE  
   b. TABLE\_PER\_CLASS  
   c. JOINED  
   d. MULTI\_TABLE  
   Answer: d. MULTI\_TABLE  
   Explanation: MULTI\_TABLE is not a valid inheritance strategy in JPA. The valid inheritance strategies are SINGLE\_TABLE, TABLE\_PER\_CLASS, and JOINED.
2. Which of the following is not a valid entity state in JPA?  
   a. NEW  
   b. MANAGED  
   c. DETACHED  
   d. PENDING  
   Answer: d. PENDING
3. What is the purpose of the @NamedQuery annotation in JPA?  
   a. It specifies a named query that can be used to retrieve entities.  
   b. It specifies the order in which entities are loaded.  
   c. It specifies the mapping of a foreign key in a relationship.  
   d. It specifies the fetch type of the relationship.  
   Answer: a. It specifies a named query that can be used to retrieve entities.  
   Explanation: The @NamedQuery annotation in JPA specifies a named query that can be used to retrieve entities. Named queries allow you to define queries that can be reused throughout your application.
4. What is the difference between a One-to-One and a Many-to-One relationship in JPA?  
   a. A One-to-One relationship is bidirectional, while a Many-to-One relationship is unidirectional.  
   b. A One-to-One relationship is unidirectional, while a Many-to-One relationship is bidirectional.  
   c. A One-to-One relationship maps to a single foreign key column, while a Many-to-One relationship maps to multiple foreign key columns.  
   d. A One-to-One relationship maps to multiple foreign key columns, while a Many-to-One relationship maps to a single foreign key column.  
   Answer: b. A One-to-One relationship is unidirectional, while a Many-to-One relationship is bidirectional.  
   Explanation: In JPA, a One-to-One relationship is unidirectional, meaning that only one entity has a reference to the other entity. In contrast, a Many-to-One relationship is bidirectional, meaning that both entities have a reference to each other.
5. Which annotation is used to specify the order of elements in a collection in JPA?  
   a. @OrderBy  
   b. @Order  
   c. @SortBy  
   d. @Sort  
   Answer: a. @OrderBy  
   Explanation: The @OrderBy annotation is used to specify the order of elements in a collection in JPA. It allows you to specify the name of a property to sort by.
6. Which of the following is not a valid mapping type for a field or property in JPA?  
   a. @Basic  
   b. @Transient  
   c. @Embeddable  
   d. @Persistent  
   Answer: d. @Persistent  
   Explanation: @Persistent is not a valid mapping type for a field or property in JPA. The valid mapping types are @Basic, @Transient, and @Embeddable.
7. What is the purpose of the @Version annotation in JPA?  
   a. It specifies the version number of an entity.  
   b. It specifies the type of the relationship between entities.  
   c. It specifies the fetch type of the relationship.  
   d. It specifies the mapping of a foreign key in a relationship.  
   Answer: a. It specifies the version number of an entity.  
   Explanation: The @Version annotation in JPA specifies the version number of an entity. This allows JPA to detect conflicts when multiple users are updating the same entity simultaneously.
8. What is the purpose of the EntityManager in JPA?  
   a. It provides a way to query the database and retrieve entities.  
   b. It manages the lifecycle of entities and performs CRUD operations.  
   c. It specifies the type of the relationship between entities.  
   d. It specifies the fetch type of the relationship.  
   Answer: b. It manages the lifecycle of entities and performs CRUD operations.  
   Explanation: The EntityManager in JPA manages the lifecycle of entities and performs CRUD (Create, Read, Update, Delete) operations. It provides a way to persist entities to the database and retrieve them.
9. Which of the following is true about CascadeType.ALL in JPA?  
   a. It specifies that all operations on the parent entity should be cascaded to the child entities.  
   b. It specifies that all operations on the child entities should be cascaded to the parent entity.  
   c. It specifies that only the CREATE operation should be cascaded to the child entities.  
   d. It specifies that only the DELETE operation should be cascaded to the child entities.  
   Answer: a. It specifies that all operations on the parent entity should be cascaded to the child entities.  
   Explanation: CascadeType.ALL in JPA specifies that all operations on the parent entity (such as create, update, and delete) should be cascaded to the child entities.
10. What is the purpose of the @GeneratedValue annotation in JPA?  
    a. It specifies the name of the generator used to generate primary key values.  
    b. It specifies the type of the relationship between entities.  
    c. It specifies the fetch type of the relationship.  
    d. It specifies the mapping of a foreign key in a relationship.  
    Answer: a. It specifies the name of the generator used to generate primary key values.  
    Explanation: The @GeneratedValue annotation in JPA specifies the name of the generator used to generate primary key values. There are different types of generators available in JPA, such as the IDENTITY, TABLE, and SEQUENCE generators.
11. Which of the following is true about the FetchType.LAZY in JPA?  
    a. It specifies that related entities should be loaded eagerly.  
    b. It specifies that related entities should be loaded lazily.  
    c. It specifies the type of the relationship between entities.  
    d. It specifies the fetch type of the relationship.  
    Answer: b. It specifies that related entities should be loaded lazily.  
    Explanation: The FetchType.LAZY in JPA specifies that related entities should be loaded lazily, meaning that the related entities are not loaded from the database until they are actually accessed.
12. Which of the following is true about the CascadeType.REMOVE in JPA?  
    a. It specifies that all operations on the parent entity should be cascaded to the child entities.  
    b. It specifies that all operations on the child entities should be cascaded to the parent entity.  
    c. It specifies that only the CREATE operation should be cascaded to the child entities.  
    d. It specifies that only the DELETE operation should be cascaded to the child entities.  
    Answer: d. It specifies that only the DELETE operation should be cascaded to the child entities.  
    Explanation: CascadeType.REMOVE in JPA specifies that only the DELETE operation should be cascaded to the child entities.
13. What is the purpose of the @IdClass annotation in JPA?  
    a. It specifies the primary key column of an entity.  
    b. It specifies the name of the generator used to generate primary key values.  
    c. It specifies the composite primary key class for an entity.  
    d. It specifies the mapping of a foreign key in a relationship.  
    Answer: c. It specifies the composite primary key class for an entity.  
    Explanation: The @IdClass annotation in JPA specifies the composite primary key class for an entity. This allows you to define a composite primary key that consists of multiple fields or properties

# Interview Questions

### **1. What do you mean by the Entity in JPA?**

The Entity is a persistent domain object which is lightweight. The Entity is the main program entity. And additional classes can also be used by it, which can further be used to maintain the state of the Entity or auxiliary classes.  
The entity class can inherit from other entity classes as well as the non-entity classes.

### **2. What is meant by the attribute of the entity class in the terminology of JPA?**

There are two elements of JPA that it can work upon. First are the properties (property) of classes that are designed like that of JavaBeans. And the second is the fields (field) or the class variables. Both these elements- property and field are the attributes of the entity class.

### **3. List some JPA requirements for entity classes.**

* An entity class should be described in the JPA's XML configuration file or annotated with the Entity.
* It should be a top-level class.
* It can neither be interface nor final class.
* It should also contain a protected or public constructor without arguments.
* The primary key should also be contained in an entity class.

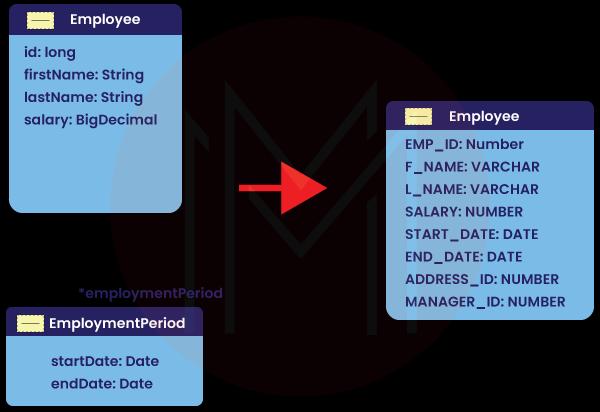
### **4. What are the types of data that the entity class attributes allow?**

The given attributes are valid in entity class:

* Strings
* Primitive types as well as their Java wrappers
* Entity types
* Enums
* Embeddable classes
* Java serializable types
* Collection types 1 to 6

### **5. What are the embeddable classes?**

An embeddable class cannot be used by itself. Both single embedded classes, as well as collections, can be contained in an entity class. We can use them as a map or key values also. Each embedded class belongs to just one object of the entity class at run time. And we can't use it for transferring data among the objects of entity classes. It serves the purpose of making definitions of common attributes for several entities.



### **6. List the valid attribute types that are included in the primary key of the entity class so that you can use them for any database.**

The list of such attributes is given below:

* Strings
* Primitive types as well as their Java wrappers
* Java.sql.Date and Java.util.Date
* BigInteger and BigDecimal

### **7. What are the requirements set by JPA for embeddable classes?**

The following requirements are set by JPA:

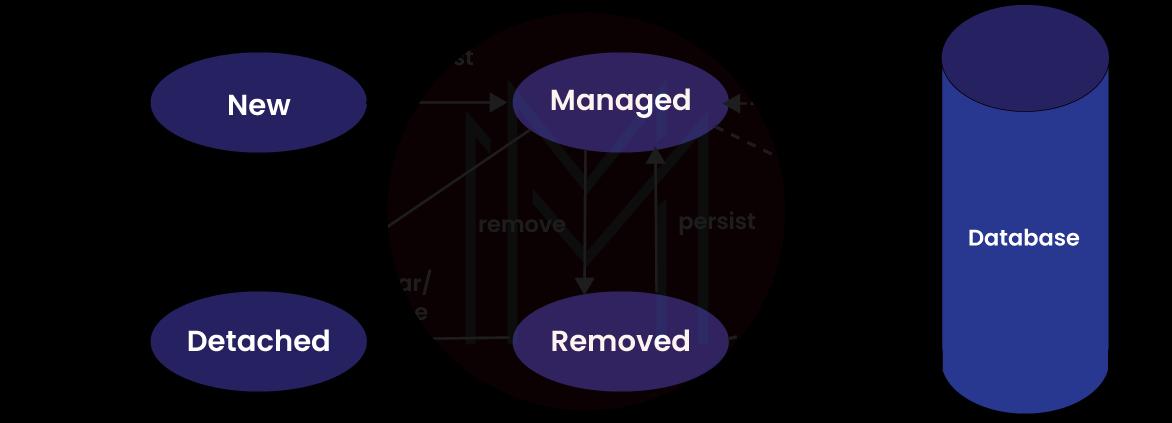
1. The embeddable class should be described in the JPA's XML configuration file or marked with embeddable annotation.
2. The embeddable classes should satisfy the same rules as the entity classes. But they need not have to contain a primary key and are marked with the annotation of the Entity.

**Check Out:**[**Top XML Interview Questions And Answers**](https://mindmajix.com/xml-interview-questions)

### **8. List the four lifecycle statuses of the life cycle of an entity instance.**

The four lifecycle statuses of an entity object are given below:

**New-** The object is created. But the primary keys have not been generated, and the object hasn't been saved yet.  
**Managed-** The object is created. Also, the primary keys have been generated, and it's managed by JPA.  
**Detached-** The object is created, but the JPA doesn't manage it.  
**Removed-** The object is created as well as managed by JPA, but it will get deleted after the transaction is committed.



### **9. What is an Entity Manager?**

An API is described by this EntityManager interface for Entity's all basic operations and also of other JPA entities.

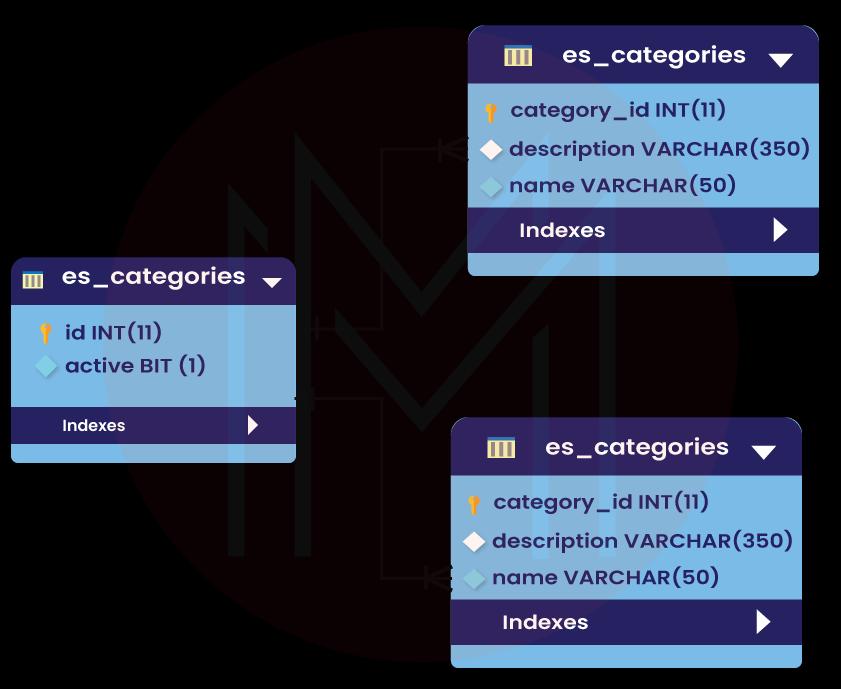
### **10. What are the main functions of Entity Manager?**

The main functions of EntityManger are given below:

* For operations on the Entity
* Data preparation
* Preparation of other entities
* Work with EntityGraph

### **11. What is meant by Mapped Superclass?**

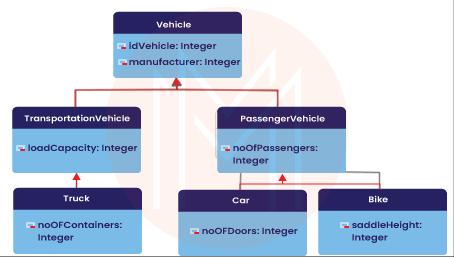
The Entity is inherited from this Mapped Superclass only. Because it is not exactly the Entity, it doesn't need to fulfil the requirements set for the Entity. We can't use this class in Query operations or EntityManager. And it must be described in an XML file and marked with the annotation of Mapped Superclass.



### **12. What three inheritance mapping strategies does the JPA describe?**

The mapping strategies are the way of working with JPA with the entity-derived classes. The three types of inheritance strategies provided are:

* Unifying strategy (the strategy of a joined subclass)
* One table for the entire inheritance hierarchy
* One table for each class



### **13. List the types of connections available between the entities.**

The following four types of connections are available:

* **OneToOne:** An object of the Entity can be associated with only one object of any other entity.
* **OneToMany:** An object of the Entity can be associated with more than one object of other another entity.
* **ManyToOne:**More than one object of an entity can be associated with another entity object.
* **ManyToMany:** More than one object can be associated with another entity's more than one object.

### **14. Which two kinds of fetch strategies are available in JPA?**

The following two types of fetch strategies are available in JPA

* **EAGER:** these fields get loaded immediately.
* **LAZY:** these fields can be only loaded on this field's first access.

### **15. Can some other embeddable class be contained in the embeddable class?**

Yes maybe.

**16. What is the Java Persistence API?**

* JPA stands for Jakarta Persistence API (JPA), formerly known as Java Persistence API.
* JPA is just a specification that facilitates object-relational mapping to manage relational data in Java applications. It provides a platform to work directly with objects instead of using SQL statements.
* JPA defines only specifications, it does not provide an implementation. JPA implementation is provided as a reference implementation by the vendors developing O/R Mappers such as [Hibernate](http://www.javaguides.net/p/jpa-tutorial-java-persistence-api.html), EclipseLink, and Apache OpenJPA.

17**. What is Object-Relational Mapping?**

The term Object/Relational Mapping refers to the technique of mapping data from an object model representation to a relational data model representation (and vice versa).

For example, let's look at the below diagram showing the mapping between the Student java model and the database relational student table.

**18 . What are the advantages of JPA?**

The advantages of JPA are given below.

The burden of interacting with the database reduces significantly by using JPA.

The user programming becomes easy by concealing the O/R mapping and database access processing.

The cost of creating the definition file is reduced by using annotations.

We can merge the applications used with other JPA providers

Using different implementations can add the features to the standard Implementation which can later be the part of JPA specification.

**19. What is the Difference Between JPA and Hibernate?**

As we know that JPA is just a specification, meaning there is no implementation. You can annotate your classes as much as you would like with JPA annotations, however, without an implementation, nothing will happen. Think of JPA as the guidelines that must be followed or an interface, while Hibernate's JPA implementation is code that meets the API as defined by the JPA specification and provides the under-the-hood functionality.

In short, [JPA](http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html) is the interface while Hibernate is the implementation.

Traditionally there have been multiple Java ORM solutions:

* [**Hibernate**](http://hibernate.org/orm/documentation/)
* [**TopLink**](http://en.wikipedia.org/wiki/TopLink)
* [**JDO**](http://en.wikipedia.org/wiki/Java_Data_Objects)
* [**Eclipse Link**](https://www.eclipse.org/eclipselink/)

Each of the above implementations defines its own mapping definition or client API. The JPA expert group gathered the best of all these tools and so they created the Java Persistence API standard.

The benefit of this is that you can swap out Hibernate's implementation of JPA for another implementation of the JPA specification.