Explore More

Subcription : Premium CDAC NOTES & MATERIAL @99

Contact to Join
Premium Group

Click to Join
Telegram Group

For More E-Notes

Join Our Community to stay Updated

TAP ON THE ICONS TO JOIN!

	codewitharrays.in freelance project available to buy contact on 8007592194	
SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySql
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySql
3	Tour and Travel management System	React+Springboot+MySql
4	Election commition of India (online Voting System)	React+Springboot+MySql
5	HomeRental Booking System	React+Springboot+MySql
6	Event Management System	React+Springboot+MySql
7	Hotel Management System	React+Springboot+MySql
8	Agriculture web Project	React+Springboot+MySql
9	AirLine Reservation System / Flight booking System	React+Springboot+MySql
10	E-commerce web Project	React+Springboot+MySql
11	Hospital Management System	React+Springboot+MySql
12	E-RTO Driving licence portal	React+Springboot+MySql
13	Transpotation Services portal	React+Springboot+MySql
14	Courier Services Portal / Courier Management System	React+Springboot+MySql
15	Online Food Delivery Portal	React+Springboot+MySql
16	Muncipal Corporation Management	React+Springboot+MySql
17	Gym Management System	React+Springboot+MySql
18	Bike/Car ental System Portal	React+Springboot+MySql
19	CharityDonation web project	React+Springboot+MySql
20	Movie Booking System	React+Springboot+MySql

	freelance_Project available to buy contact on 8007592194	
21	Job Portal web project	React+Springboot+MySql
22	LIC Insurance Portal	React+Springboot+MySql
23	Employee Management System	React+Springboot+MySql
24	Payroll Management System	React+Springboot+MySql
25	RealEstate Property Project	React+Springboot+MySql
26	Marriage Hall Booking Project	React+Springboot+MySql
27	Online Student Management portal	React+Springboot+MySql
28	Resturant management System	React+Springboot+MySql
29	Solar Management Project	React+Springboot+MySql
30	OneStepService LinkLabourContractor	React+Springboot+MySql
31	Vehical Service Center Portal	React+Springboot+MySql
32	E-wallet Banking Project	React+Springboot+MySql
33	Blogg Application Project	React+Springboot+MySql
34	Car Parking booking Project	React+Springboot+MySql
35	OLA Cab Booking Portal	React+NextJs+Springboot+MySql
36	Society management Portal	React+Springboot+MySql
37	E-College Portal	React+Springboot+MySql
38	FoodWaste Management Donate System	React+Springboot+MySql
39	Sports Ground Booking	React+Springboot+MySql
40	BloodBank mangement System	React+Springboot+MySql

41	Bus Tickit Booking Project	React+Springboot+MySql
42	Fruite Delivery Project	React+Springboot+MySql
43	Woodworks Bed Shop	React+Springboot+MySql
44	Online Dairy Product sell Project	React+Springboot+MySql
45	Online E-Pharma medicine sell Project	React+Springboot+MySql
46	FarmerMarketplace Web Project	React+Springboot+MySql
47	Online Cloth Store Project	React+Springboot+MySql
48	Train Ticket Booking Project	React+Springboot+MySql
49	Quizz Application Project	JSP+Springboot+MySql
50	Hotel Room Booking Project	React+Springboot+MySql
51	Online Crime Reporting Portal Project	React+Springboot+MySql
52	Online Child Adoption Portal Project	React+Springboot+MySql
53	online Pizza Delivery System Project	React+Springboot+MySql
54	Online Social Complaint Portal Project	React+Springboot+MySql
55	Electric Vehical management system Project	React+Springboot+MySql
56	Online mess / Tiffin management System Project	React+Springboot+MySql
57		React+Springboot+MySql
58		React+Springboot+MySql
59		React+Springboot+MySql
60		React+Springboot+MySql

Spring Boot + React JS + MySQL Project List

Sr.No	Project Name	YouTube Link
1	Online E-Learning Hub Platform Project	https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW
2	PG Mate / Room sharing/Flat sharing	https://youtu.be/4P9clHg3wvk?si=4uEsi0962CG6Xodp
3	Tour and Travel System Project Version 1.0	https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12
4	Marriage Hall Booking	https://youtu.be/VXz0kZQi5to?si=llOS-QG3TpAFP5k7
5	Ecommerce Shopping project	https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq
6	Bike Rental System Project	https://youtu.be/FIzsAmIBCbk?si=7ujQTJqEgkQ8ju2H
7	Multi-Restaurant management system	https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB
8	Hospital management system Project	https://youtu.be/IynIouBZvY4?si=CXzQs3BsRkjKhZCw
9	Municipal Corporation system Project	https://youtu.be/cVMx9NVyI4I?si=qX0oQt-GT-LR_5jF
10	Tour and Travel System Project version 2.0	https://youtu.be/_4u0mB9mHXE?si=gDiAhKBowi2gNUKZ

Sr.No	Project Name	YouTube Link
11	Tour and Travel System Project version 3.0	https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug
12	Gym Management system Project	https://youtu.be/J8_7Zrkg7ag?si=LcxV51ynfUB7OptX
13	Online Driving License system Project	https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn
14	Online Flight Booking system Project	https://youtu.be/m755rOwdk8U?si=HURvAY2VnizIyJlh
15	Employee management system project	https://youtu.be/lD1iE3W_GRw?si=Y_jv1xV_BljhrD0H
16	Online student school or college portal	https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD
17	Online movie booking system project	https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSlSm
18	Online Pizza Delivery system project	https://youtu.be/Tp3izreZ458?si=8eWAOzA8SVdNwlyM
19	Online Crime Reporting system Project	https://youtu.be/0UlzReSk9tQ?si=6vN0e70TVY1GOwPO
20	Online Children Adoption Project	https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N

MCQ on JPA and Hibernate

Q#1. 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

Explanation: ORM stands for Object Relational Mapping. It is a technique that offers developers to map objects from the application code to relational database tables.

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

```
(a) Hibernate
(b) TopLink
(c) MyBatis
(d) Gradle
```


Answer: d) Gradle

Explanation: Hibernate, TopLink, and MyBatis (formally known as iBatis) are the ORM frameworks. Gradle is a build tool like Maven.

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

Explanation: Hibernate simplifies database querying by providing an object-oriented approach to interact with the database. It reduces the need for writing complicated SQL queries and controls the database operations behind the scenes.

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 CascadeType should he use?

```
(a) CascadeType.REMOVE
(b) CascadeType.MERGE
(c) CascadeType.DELETE
(d) CascadeType.REFRESH
```


Answer: c) CascadeType.DELETE

Explanation: CascadeType.REMOVE and CascadeType.DELETE both propagate the delete operation from parent to child entity. CascadeType.REMOVE is specific to JPA, whereas CascadeType.DELETE is specific to Hibernate.

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

Explanation: Based on the provided mapping metadata, Hibernate can automatically produce the database schema. It eliminates the need to create database tables manually and helps in preserving the database structure.

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

Explanation: JPA stands for Java Persistence API. It is a Java specification that offers a standardized way to interact with databases using ORM technology.

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

Explanation: In JPA, the EntityManager does the job of Session Factory. Session Factory is a component specific to Hibernate, and not a part of the JPA specification.

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

Explanation: The EntityManager is responsible for managing entity persistence and retrieval operations in JPA. It gives methods to create, read, update, and delete entities from the database.

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.

Explanation: The @Table annotation is used to create a user defined table in JPA with the help of 'name' attribute and used in combination with the @Entity annotation.

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

Explanation: A persistence unit in JPA is a logical group of related entities. It defines the set of entities that can be managed as a group.

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 indep
(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.

Explanation: The JPQL (Jakarta Persistence Query Language: Formerly Java Persistence Query Language) is an object-oriented query language. It is used to perform database operations on persistent objects. JPQL uses entity object model to operate the SQL queries rather than database tables. JPA converts JPQL into SQL. It can also 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 associ
(b) A many-to-many relationship exists between customers and products if 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.

Explanation: Many-to-many relationships are not favorably recommended in a relational database because it can create many problems, such as Data redundancy, data insertion, deletion, and updating difficulties.

Q#13. Which of the following is supported by Hibernate?

```
(a) HQL
(b) SQL
(c) JPQL
(d) All of the above
```

Answer: d) All of the above

Explanation: Hibernate supports several query languages. HQL (Hibernate Query Language) is specific to Hibernate. SQL (Structured Query Language) is the standard database query language.

JPQL (Jakarta Persistence Query Language) is the query language defined in the JPA.

Q#14. Which is incorrect about the owning side in the context of ORM technique?

```
(a) It is one of the two sides of a bidirectional relationship.
(b) The owning side initiates the creation of the relationship to the database.
(c) Generally, this is the side where the foreign key resides.
(d) Generally, this is the side where the primary key resides.
```

Answer: d) Generally, this is the side where the primary key resides.

Explanation: In fact, owning side and inverse side are the technical terminologies of the ORM technology. They are not the concepts of participating entities. In fact, they are the two sides of a bidirectional relationship. The owning side initiates the creation of the relationship to the database. Generally, this is the side where the foreign key resides. For more details on this, kindly go through the topic 'owning side & inverse side'.

Q#15. What is the intent of the 'mappedBy' attribute in a bidirectional association mapping?

```
(a) It specifies the mapped column name
(b) It defines the target entity class
(c) It determines the primary key relationship
(d) It contains the name of the association-field on the owning side
```


Answer: d) It represents the owning side of the relationship

Explanation: In a bidirectional association mapping, the mappedBy attribute is used to indicate the name of the association-field on the owning side. It specifies the property in the target entity that maps back to the owning entity.

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

```
(a) =
(b) :=
```

```
(c) ?
(d) :
```

Answer: d) :

Explanation: The colon (:) operator is used for named parameter binding in JPQL. When constructing a JPQL query, we can use named parameters prefixed with a colon to represent values that will be provided at runtime.

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.

Explanation: The Hibernate SessionFactory is responsible for the creation of Session objects. It is used to create Hibernate Session objects that can perform CRUD-based operations.

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 en
(c) Hibernate supports Java language features beyond the JPA.
(d) Hibernate and JPA are two different ORM frameworks for different programming la
```

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

Explanation: Hibernate is a particular implementation of the JPA specification. JPA provides a standard set of APIs and annotations for object-relational mapping, while Hibernate is one of the implementations that follow to this specification.

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

Explanation: The @Id annotation is used to mark a field as the primary key in JPA. It indicates that the corresponding attribute is the unique identifier for the entity.

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

Explanation: A standard primary key consists of a single Java field which maps to a single table column. Collection type primary key is not possible.

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
(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.

Explanation: By default, @OneToMany and @ManyToMany associations use the FetchType.LAZY strategy while the @OneToOne and @ManyToOne 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

Explanation: The @Table annotation is used to specify the name of the table associated with an entity in Hibernate. It allows us to customize the table name and other properties of the 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

Explanation: The @GeneratedValue annotation in Hibernate is used to specify the strategy for primary key generation. It allows you to define how the primary key values for entities should be automatically generated.

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

Explanation: The Hibernate dialect is used to specify the database-specific SQL syntax and behavior. It provides the necessary translations and adaptations to generate the appropriate SQL statements for a particular database system.

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

Explanation: One of the disadvantages of using Hibernate is the possible performance overhead due to the object-relational mapping. The additional processing required for mapping objects to database tables can introduce some performance overhead.

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

Explanation: To create a native SQL query in Hibernate, we can use the createSQLQuery() method. This method allows you to create SQL queries directly without any Hibernate-specific query language.

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.

Explanation: The @EmbeddedId annotation is used in Hibernate to specify the primary key for an entity using an embedded object. It allows you to define a composite primary key by embedding multiple attributes within a separate class.

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

Explanation: To configure a second-level cache for an entity in Hibernate, we can use the @org.hibernate.annotations.Cache annotation. This annotation allows us to specify the caching settings for the entity, such as the cache region and cache concurrency strategy etc.

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’.

Explanation: If you are still using java.util.Date or java.util.Calendar as your field types, you need to annotate the field with @Temporal. Using this annotation, you can define if the attribute shall be mapped as an SQL DATE, TIME, or TIMESTAMP.

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

Explanation: The find() method of the EntityManager in JPA is used to retrieve an entity by its primary key. It takes the entity class as the first argument and the primary key value as the second arguments and returns the corresponding entity if found.

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.

Explanation: The CascadeType.PERSIST option in JPA specifies that entity insertion should be cascaded to related entities. It means that when an entity is persisted, any associated entities will also be persisted.

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

Explanation: The @JoinColumn annotation in JPA is used to specify the column name for joining an entity association or element collection. It allows us to customize the name of the column that holds the foreign key reference.

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

Explanation: To enable automatic table creation for entities in Hibernate, we can set the hibernate.hbm2ddl.auto property to "create" in the configuration. The 'create' option instructs Hibernate to drop the database schema and recreate it 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 provide native query support.

Explanation: HQL and JPQL provide native query support that isn't possible with the Criteria queries.

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.

Explanation: Hibernate Query Language (HQL) is an object-oriented query language. Unlike SQL, which operates on tables and columns, HQL works with persistent objects and their properties. HQL queries are translated by Hibernate into traditional SQL queries

JPA (Java Persistence API) Multiple-Choice Questions (MCQs)

JPA (Java Persistence API) 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 (Java Persistence API) MCQs: This section contains multiple-choice questions and answers on the various topics of JPA (Java Persistence API). Practice these MCQs to test and enhance your skills on JPA (Java Persistence API).

List of JPA (Java Persistence API) MCQs

1. JPA stands for ___?

A. Java persistence API
B. Java programming API
C. Java persistence Arguments

Answer: A) Java persistence API

Explanation:

JPA stands for Java persistence API.

Discuss this Question

---

2. JPA is used for ___?

A. For data access, persistence, and management in Java applications
B. For testing java applications
C. For creating runtime API
D. All of the above

Answer: A) For data access, persistence, and management in Java applications

Explanation:

JPA is used for data access, persistence, and management in Java applications.

Discuss this Question

---

3. A database table is represented by an ___, which is a Java class.

A. Persistence
B. Entity
C. Primary key

D. Relationships

Answer: B) Entity

Explanation:

A database table is represented by an entity, which is a Java class.

Discuss this Question

---

4. ____ is a class that allows you to persist, update, and delete entities.

A. Persistence
B. Entity
C. EntityManager
D. Bias

Answer: C) EntityManager

Explanation:

EntityManager is a class that allows you to persist, update, and delete entities.

Discuss this Question

---

5. Which of the following is an entity's unique identifier?

A. Transactions
B. Entity
C. Primary key
D. Relationships

Answer: C) Primary key

Explanation:

A primary key is an entity's unique identifier.

Discuss this Question

---

6. Is JPA an open-source API?

A. Yes
B. No

Answer: A) Yes

Explanation:

Yes, JPA is an open-source API.

Discuss this Question

---

7. In a relationship mapping, which of the following annotations is used to indicate the join column?

A. @Join
B. @Column
C. @JoinColumn
D. All of the above

Answer: C) @JoinColumn

Explanation:

@JoinColumn annotation is used to specify the join column in relationship mapping.

Discuss this Question

---

8. ____ annotation is used to specify a Query by providing a static name?

A. @list
B. @NamedQuery
C. @NamedQueries
D. @querieslist

Answer: B) @NamedQuery

Explanation:

@NamedQuery annotation is used to specify a Query by providing a static name.

Discuss this Question

---

9. Which of the following annotations is used to define a list of named queries?

A. @list
B. @NamedQuery
C. @NamedQueries
D. @querieslist

Answer: C) @NamedQueries

Explanation:

@NamedQueries annotation is used for specifying the list of named queries.

Discuss this Question

---

10. Which of the following annotations is used to indicate an entity's primary key field?

A. @pk
B. @primary
C. @key
D. @id

Answer: D) @id

Explanation:

@id annotation is used to specify the primary key field of an entity.

Discuss this Question

---

11. ___ is a programming capability that allows data to be converted from object type to relational type and vice versa.

A. ORM
B. JPA
C. JRE
D. Java RMI

Answer: A) ORM

Explanation:

ORM is a programming capability that allows data to be converted from object type to relational type and vice versa.

Discuss this Question

---

12. Which of the following annotations defines a class as an entity?

A. @table
B. @entitymanager
C. @entityfactory
D. @entity

Answer: D) @entity

Explanation:

@entity annotation is used to define a class as an entity.

Discuss this Question

---

13. What do you mean by ORM?

A. Object relationship mapping
B. Object-oriented relationship map
C. Originator relationship mapper
D. Object-relational mapper

Answer: D) Object-relational mapper

Explanation:

ORM stands for object-relational mapper.

Discuss this Question

---

14. What is the relationship between EntityManager and entity?

A. One-to-one
B. Many-to-one
C. One-to-many

Answer: C) One-to-many

Explanation:

The relationship between EntityManager and entity is one-to-many.

Discuss this Question

---

15. What is the relationship between EntityManager and Query?

A. One-to-one
B. Many-to-one
C. One-to-many

Answer: C) One-to-many

Explanation:

The relationship between EntityManager and Query is one-to-many.

Discuss this Question

---

16. EntityManager and EntityTransaction have a ___- relationship?

A. One-to-one
B. Many-to-one
C. One-to-many

Answer: A) One-to-one

Explanation:

EntityManager and EntityTransaction have a one-to-one relationship.

Discuss this Question

---

17. EntityManagerFactory and EntityManager have which kind of relationship?

A. One-to-one
B. Many-to-one
C. One-to-many

Answer: C) One-to-many

Explanation:

EntityManagerFactory and EntityManager have a one-to-many relationship.

Discuss this Question

---

18. ____ is a factory class that generates EntityManager instances.

A. Entity
B. EntityManagerFactory
C. EntityManager
D. Persistence unit

Answer: B) EntityManagerFactory

Explanation:

EntityManagerFactory is a factory class that generates EntityManager instances.

Discuss this Question

---

19. The ____ serves as a logical container for a collection of entity classes.

A. Entity
B. EntityManagerFactory
C. EntityManager
D. Persistence unit

Answer: D) Persistence unit

Explanation:

The persistence unit serves as a logical container for a collection of entity classes.

Discuss this Question

---

20. JPA is based upon which of the following architecture?

A. MVC
B. Layered architecture
C. MVVM
D. Client-server architecture

Answer: B) Layered architecture

Explanation:

JPA is based upon layered architecture.

Discuss this Question

---

21. JPA supports how many different inheritance strategies?

A. 5
B. 8
C. 4
D. 3

Answer: D) 3

Explanation:

JPA supports three types of inheritance strategies:SINGLE_TABLE, JOINED_TABLE, and TABLE_PER_CONCRETE_CLASS.

Discuss this Question

---

22. Does JPA supports object-oriented concept?

A. Yes
B. No

Answer: A) Yes

Explanation:

Yes, JPA support the object-oriented concept.

Discuss this Question

---

23. By default, the connected entities are not retrieved alongside the primary item under the ____ strategy.

A. Eager fetch
B. Lazy fetch.

Answer: B) Lazy fetch.

Explanation:

By default, the connected entities are not retrieved alongside the primary item under the Lazy Fetch strategy.

Discuss this Question

---

24. Which of the following is the default fetch strategy in JPA?

A. Eager fetch
B. Lazy fetch.

Answer: A) Eager fetch

Explanation:

Eager Fetch is the default fetch strategy in JPA.

Discuss this Question

---

25. How ways are there to fetch records from the database?

A. 2
B. 3
C. 4
D. 5

Answer: A) 2

Explanation:

There are two methods for retrieving records from a database: eager fetch and lazy fetch.

Discuss this Question

---

26. ___ is used to create queries against entities to store in a relational database.

A. JPA API
B. Persistence Objects
C. ORM
D. JPQL

Answer: D) JPQL

Explanation:

JPQL is used to create queries against entities to store in a relational database.

Discuss this Question

---

27. What is JPQL?

A. Java Programming Querying Language
B. Java Programs Query Language
C. Java persistence Query Language

Answer: C) Java persistence Query Language

Explanation:

JPQL stands for Java Persistence Query Language.

Discuss this Question

---

28. Any property's Setter method should begin with ___?

A. with Captial lettered 'Set'
B. with small lettered 'set'
C. Any letter

Answer: B) with small lettered 'set'

Explanation:

Any property's setter method should begin with a small lettered 'set'.

Discuss this Question

---

29. Any property's getter method should begin with ___?

A. with Captial lettered 'Get'
B. with small lettered 'get'
C. Any letter

Answer: B) with small lettered 'get'

Explanation:

Any property's getter method should begin with a small lettered 'get'.

Discuss this Question

---

30. The Java class encapsulates instance data and behaviors into a single unit known as an ____.

A. Object
B. Instance
C. Persistence Class
D. Java RMI

Answer: A) Object

Explanation:

The Java class encapsulates instance data and behaviors into a single unit known as an object.

Discuss this Question

---

31. The related entities are obtained alongside the primary item in a single query under the ____ strategy.

A. Eager Fetch
B. Lazy Fetch

Answer: A) Eager Fetch

Explanation:

The related entities are obtained alongside the primary item in a single query under the Eager Fetch strategy.

Discuss this Question

---

32. Which of the following inheritance strategy maps all of an entity's subclasses to the same table in the database?

A. Table per class strategy
B. Joined table strategy
C. Single table strategy

Answer: C) Single table strategy

Explanation:

Single table Strategy maps all of an entity's subclasses to the same table in the database.

Discuss this Question

---

33. The ____ strategy involves creating a table for each sub entity.

A. Table per class strategy
B. Joined table strategy
C. Single table strategy

Answer: A) Table per class strategy

Explanation:

The table-per-class strategy involves creating a table for each sub-entity.

Discuss this Question

---

34. In what type of relationship is one entity connected with only one instance of another entity?

A. @ManyToOne Relation
B. @OneToMany Relation
C. @OneToOne Relation
D. @ManyToMany Relation

Answer: C) @OneToOne Relation

Explanation:

One entity is connected with one and only one instance of another entity in a one-to-one relationship.

Discuss this Question

---

35. Which of the following relationships links one entity with one or more instances of another entity?

A. @ManyToOne Relation
B. @OneToMany Relation
C. @OneToOne Relation
D. @ManyToMany Relation

Answer: B) @OneToMany Relation

Explanation:

One entity is connected with one or more instances of another entity in a one-to-many relationship.

Discuss this Question

36. Several instances of one entity are connected with many instances of another entity in a ____ relationship.

A. @ManyToOne Relation
B. @OneToMany Relation
C. @OneToOne Relation
D. @ManyToMany Relation

Answer: D) @ManyToMany Relation

Explanation:

Several instances of one entity are connected with many instances of another entity in a many-to-many relationship.

Discuss this Question

37. How many types of caching does JPA support?

A. 5
B. 4
C. 2
D. 3

Answer: C) 2

Explanation:

JPA provides two types of caching: first-level cache and second-level cache.

Discuss this Question

38. Which of the following type of cache is also known as the entity manager cache?

A. First-level cache
B. Second-level cache.

Answer: A) First-level cache

Explanation:

The first-level cache is also known as the entity manager cache.

Discuss this Question

39. Which of the following type of cache is active by default in JPA and cannot be disabled?

A. First-level cache
B. Second-level cache.

Answer: A) First-level cache

Explanation:

JPA's first-level cache is enabled by default and cannot be deactivated.

Discuss this Question

---

40. A superclass is defined as an object without a matching database table in ____ Inheritance, and its attributes and mappings are inherited by its subclasses.

A. Table per class strategy
B. Joined table strategy
C. Single table strategy
D. Mapped Superclass

Answer: D) Mapped Superclass

Explanation:

A superclass is defined as an object without a matching database table in Mapped Superclass Inheritance, and its attributes and mappings are inherited by its subclasses.

https://www.youtube.com/@codewitharrays

https://www.instagram.com/codewitharrays/

https://t.me/codewitharrays Group Link: https://t.me/ccee2025notes

+91 8007592194 +91 9284926333

codewitharrays@gmail.com

https://codewitharrays.in/project