1. What is JPA?  
Answer: JPA stands for Java Persistence API. This is a Java specification used to persist data between the relational database and Java objects. It is the bridge between object-oriented domain models and relational databases.  
  
2. What are the advantages of JPA?  
Answer:

* JPA is user-friendly
* It helps to reduce the burden of interacting with databases
* Through annotation, it helps in reducing the cost of creating a definition file.
* Useful in merging applications
* Makes user programming easy

3. What are the different types of entity mapping?  
Answer:

* One-to-one mapping
* One-to-many mapping
* Many-to-one mapping
* Many-to-many mapping

4. Why is an interface not a class?  
Answer: The reason why interface is not a class it is because it does not contain concrete methods. The only thing it can contain is [abstract method](https://javarevisited.blogspot.com/2022/08/can-we-declare-abstract-class-or-method.html)s.

5. What is an entity?  
Answer: An entity refers to a group of states associated together in a single unit.

6. Discuss the important predefined repository interfaces and classes in Spring Data JPA  
Answer:

* Repository – this is a top-level interface defined in Spring Data Hierarchy
* CrudRepository – CrudRepository interface extends Repository interface and it also provides methods that help to perform CRUD operation.
* PagingAndSortingRepository – this interface extends CrudRepository interface and provides additional methods to retrieve entities using the pagination and sorting.
* QueryByExampleExecutor – this interface is used to execute Query by example.
* [JpaRepository](https://javarevisited.blogspot.com/2021/10/what-is-spring-data-repository.html) – JpaRepository interface extends PagingAndSortingRepository and QueryByExampleExecutor interface. It also helps in providing some additional batch methods.
* SimpleJpaRepository – SimpleJpaRepository is basically the implementation class of the CrudRepository interface.
* QueryDslRepository – this is a class.

7. What is Object Relational Mapping (ORM)?  
Answer: Object Relational Mapping (ORM) is defined as a mechanism responsible for maintaining the relationship between object oriented data structures and relational tables in a database. Common ORM frameworks are JPOX, ORMLite, Toplink, [Hibernate](https://javarevisited.blogspot.com/2018/01/top-5-hibernate-and-jpa-courses-for-java-programmers-learn-online.html)and iBatis.

8. What are the differences between JpaRepository and CrudRepository?  
Answer:

|  |  |
| --- | --- |
| JpaRepository | CrudRepository |
| It extends QueryByExampleExecutor and PagingAndSortingRepository interface | it extends Repository interface |
| The saveAll method of JpaRepository returns list | The saveAll method of CrudRepository returns Iterable |
| Provides additional methods like deleteInBatch(), flush() and others | Provides methods to perform CRUD operations |

9. What is the difference between getOne() and findById()?  
Answer: getOne() is available in JpaRepository while findById() is available in CrudRepository.  
  
10. What methods of CrudRepository apply in performing CRUD operations?  
Answer:

* existsById – used to check if an entity of a given id already exists in the database.
* findAll() – find all entity of particular type.
* save(S entity) – it is used to save a single entity at a time.
* findById – used to get entity by id.
* findAllById – returns al entities of given ids.
* deleteById – used to delete the entity basing on id.
* count() – it returns the given number of entities.
* deleteAll() – delete all entities.

11. What does the @Entity annotation do?  
Answer: the @Entity annotation indicates a class represents a relational table in the database. The JPA specification includes any class marked with @Entity in the persistence setup.

12. What does the @EnableJpaRepositories annotation do?  
Answer: this annotation enables the automatic generation of JPA repositories. Any class which implements CrudRepository interface will generate a repository when this annotation is present.  
  
13. What does the @Query annotation do? (answer)  
Answer: The @Query annotation allows you to define a Spring Data Repository method with custom SQL. With the use of [@Query](https://javarevisited.blogspot.com/2021/09/spring-data-jpa-query-example-tutorial.html), you can map Spring Data Repository methods to actual SQL statements.  
  
14. What does the @Id annotation do?  
Answer: The @Id annotation marks a field as the primary key for that particular table. It is a unique identifier for each entry in the table. @Id annotation is typically used with @GeneratedValue to automatically generate a unique id for each entry in the table.

15. What is the difference between Hibernate and Spring Data JPA?  
Answer: Hibernate is a JPA implementation while Spring Data JPA is a JPA Data Access Abstraction.

16. How is a custom repository created in Spring data JPA?  
Answer: A custom repository is created when it is extended to any of the following interfaces: CrudRepository, JpaRepository, Repository, QueryByExampleRepository and PagingAndSortingRepository.

17. What does the @GeneratedValue annotation do?  
Answer: The @GeneratedValue annotation is used to specify the primary key generation strategy to use. If by any chance the strategy is not specified by default AUTO will be used.

18. What is the difference between FetchType.Eager and FetchType.Lazy?  
Answer: FetchType attribute indicates how whether records will be eagerly or lazily loaded from the database. When the records are eagerly loaded, JPA returns these objects regardless of whether they are accessed by the client or not. When records are lazily loaded the actual objects are only retrieved when directly accessed. This helps in saving memory and processing when appropriate.

19. What does @Column annotation do?

Answer: The @Column annotation is utilized to designate the details of the column to which a field or property will be mapped.

20. What does the @EntityScan annotation do?  
Answer: If the entity classes are not placed in the main application package or its sub package, then it is required to declare the package in the main configuration class with @EntityScan annotation.