1. 3-tier architecture?

=> - Design to interface

Contract based on specification

- Adaptability

- Scalability

- Improved testability

- Separate of concern

- Domain mode

Central

2. Different between IoC and DI?

- IoC – Inversion of Control. Mechanism of control. The way to control code with configuration without worry in business code

business code

Ioc Container Configuration Injection



- DI – Dependency Injection

3. Three advantage of DI

- loose couples

- simplifies implementation

- support adaptability, scalability, testability

4. What is POJO? How to use in spring

Plain Old Java Object.

* Private filed
* Setter and getter
* No dependency (because spring is framework)

Quiz 2:

1. What is ORM? What’s the value?

* It’s gate way between object and relational database.
* Developer don’t care about connection database and the field in query. A lot of generate for you. It’s mismatch. The syntax to get something in short.

1. What’s persistence context? Explain it in terms of transient, detected, removed object

* Other name is session per request. Session is start time and end time.
* For example, is session in class. When phone call, open the context with each other for connected with open time and close it.
* Transient: - OO doesn’t go to db. It’s tempory and not value
* Detected: - Something is connected with other something
* Removed: -

1. What does a Merge do? How does it work?

* Merge == update.

Quiz 3:

1. What is Impedance Mismatch? Explain by using Foreign Keys as an example.

* How to deal with 2 different two technologies between ORM and relational database
* In term of syntax
* Association and relation problem need to deal.
* Foreign key : who own relationship in OO environment relational model. Example: Member class and Address class. Who is the owner?

1. Explain the 2 fetch types.

* FetchType.ENGER:
* FetchType.LAZY:

1. What is a Join Table?
2. Explain OrphanRemoval.

– Explain with composite relationship

Orphan Removal – when a target entity in one-to-one or one-to-many relationship is removed from the relationship, it is often desirable to cascade the remove operation to the target entity.  
Example: if an order has many line items and one of them is removed from the order, the removed line item is considered an orphan. If orphanRemoval is set to true, the line item entity will be deleted when the line item is removed from the order.

Orphan Removal removes corresponding child when you remove it from the relationships. So, if you delete 1 photo from user.getPhotos() collection, JPA automatically removes that photo from database too

Quiz 4:

Question 1 of 3

Give an example of a parameterized JPL query

|  |  |
| --- | --- |
| |  | | --- | | public Member findByMemberNumber(Integer number) {    Query query = entityManager.createQuery("select m from Member m where m.memberNumber =:number");  return (Member) query.setParameter("number", number).getSingleResult();  } | |
| Question 2 of 3  Version-Based Optimistic Concurrency [Locking] : How does it relate to isolation levels.  How it is implemented in JPA.  Isolation level put repeatable read.  Detach object   |  | | --- | |  | |
| Question 3 of 3  What is the Criteria API? What is a good use case for it? |

Mid Exam :

Lesson1: 7, 9, 11

Lesson1 spring: 3: spring core technologies

* 27 Spring compent annation

Lesson2 a ORM:

* 9 - basic function of ORM
* 10 – Objectional Relational Map the Domain Model in a Relationship world
* 11
* 13 ORM Use Case
* 14 what is reason ORM
* 24 – Spring ORM support
* 25 – How it works
* 27 – JPQL Data object queries (need to remember syntax)

Lesson2 b - :

* @GenerateValue

Lesson3:

Lesson4 a: - forget

B: -

* JPQL (main query don’t care)
* Explain the join

@MemberRole : seem like sub query

Lesson5:

* 2 – Performance consider
* Customized Fetching Strategies