CS544
Enterprise Architecture
Midterm May 2017

Name_____

Student ID _____

NOTE: This material is private and confidential. It is the property of MUM and is not to be disseminated.

1. [15 points] **Circle w**hich of the following is TRUE/FALSE concerning Spring Transaction Management:

   T F Every interaction with an RDBMS requires a transaction whether a READ or a WRITE. Without a Transaction Management capability like Spring's, DB operations would fail.

   EXPLAIN**:**_____

   T F Spring Transaction Management is based on a logical unit of work.

   EXPLAIN:_____

   T F Using Spring Transaction Management with JPA requires a Persistence Context.

   EXPLAIN:_____

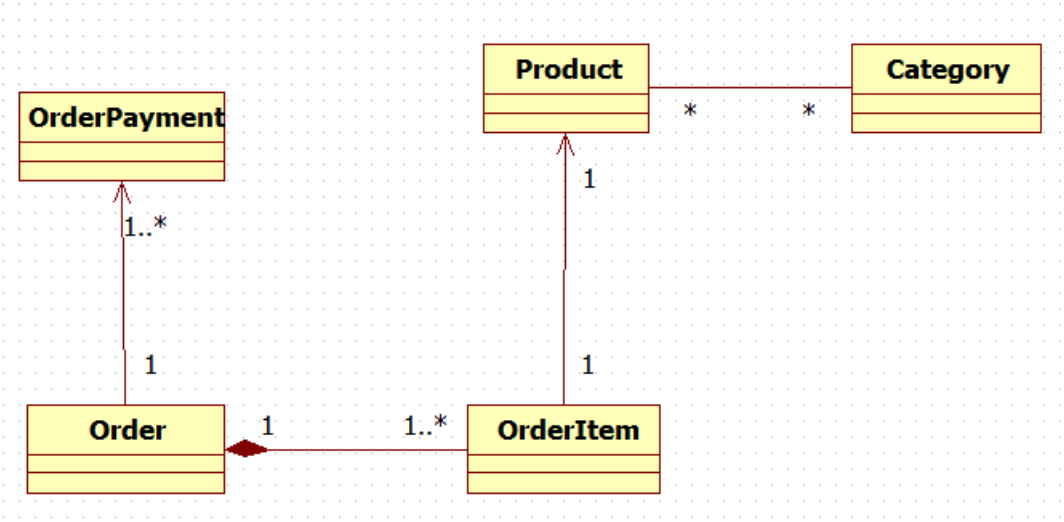
   T F Spring @Transaction has no built-in metadata for managing any of the DB ACID properties.

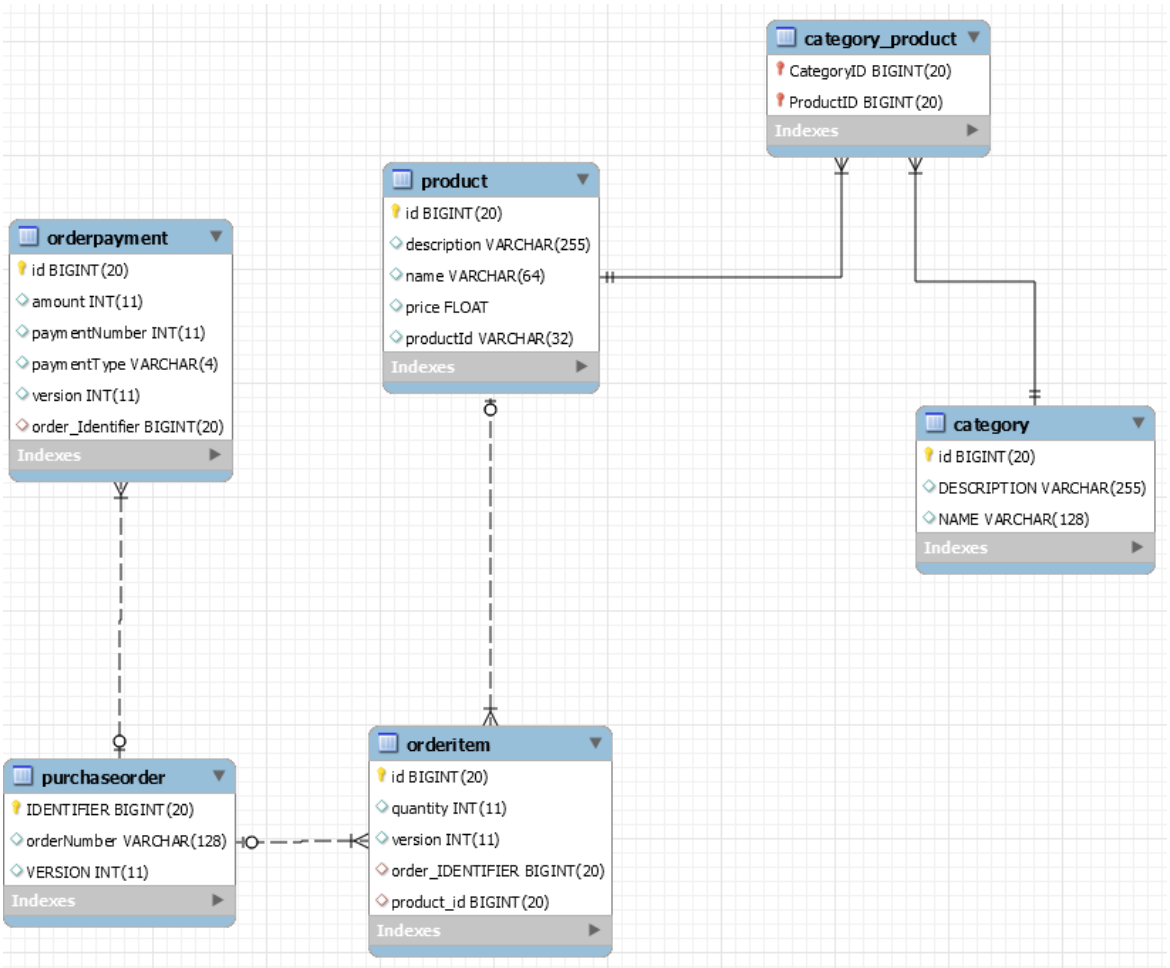
   EXPLAIN:_____

   T F Spring Declarative Transaction Management requires little or no application code related to transaction management.

   EXPLAIN:_____

2. [20 points] Annotate the Domain Objects based on the Domain Model and Entity Relationship Diagram provided. NOTE: All the fields are not listed. Only annotate the fields that are listed.





Product.java

```
public class Product {

private Long id = null;

private String Name;

private String description;

private String productId;

private Float price;

private Set<Category> categories;
```

Order.java

```
public class Order {

    private Long id = null;

    private int version = 0;

    private String orderNumber;


   private Set(OrderItem> items;


   private Set<OrderPayment> payments;
```

Category.java

```
public class Category {

    private Long id = null;

    private String Name;

    private String description;

    private Set<Product> products;
```

OrderItem.java

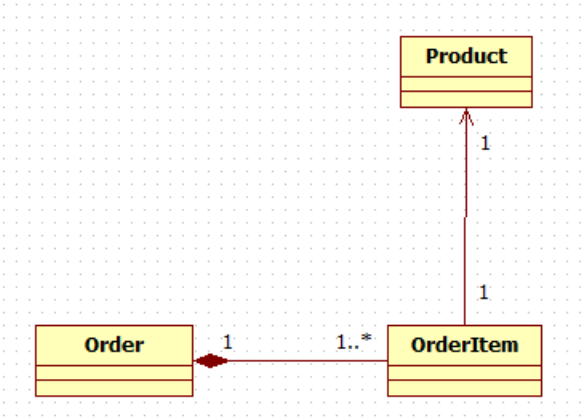
```
public class OrderItem {

    private Long id = null;

    private quantity;

    private Order order;

    private Product product;
```

OrderPayment.java

```
public  class OrderPayment  {

    private Long id = null;

    private String paymentType;

    private Integer amount;
```

3. [15 points] The reason for an ORM is because object models and relational models do not work very well together. Describe what is known as the Object-Relational Impedance mismatch. Give specific examples of the problems that arise from the mismatch.

4. [15 points]] For the following relationships implement a Join fetch of all Orders with their Order Item collection



What performance problem does the Join fetch address? Give details.
What performance problem does it cause? Give details.
What can be done to “clean up” the data returned by the fetch?

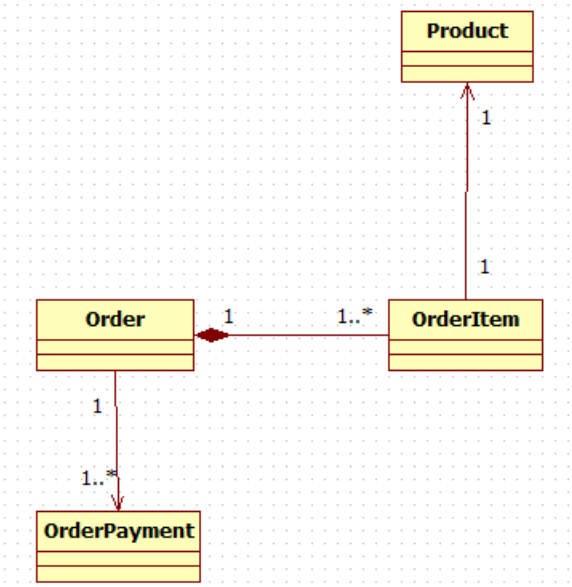
5. [15 points] Explain the concept of ORM caching. Include a discussion of :
   - First level relate to Persistence Context; Fetch Strategy
   - Second level
     - Read-only - read-write
     - Second-level .vs. query
     - When do you decide to use a second level cache?

   Be specific. Give examples. Diagrams are good.

6. [15 points] Implement a parameterized JQPL query with this signature:

```
public List<Product> findByAmountRangeAndQuantity(Integer minPayment,
                                                  Integer maxPayment,
                                                  Integer quantity)
```

The query looks up all Product[s] where the Order Item quantity is greater than the supplied quantity and the Order Payment Amount is within the supplied parameters.



The Query should be a parameterized query. Also identify all the classes in the specific packages that need to be modified to adhere to the N-Tier architecture convention.