**Object Relational Mapping with JPA**

1. ***Explain the rationale behind the topic Object Relational Mapping and the Pros and Cons in using ORM.***Object relational mapping is a technology to extract and persist data between the IDE and a database. In JPQL the tables, rows and columns from the database, will often refer as: List<Object>, objects and attributes.

* **Pros**

Reduces the amount of mapping work that has to be done.  
Avoids the JDBC and SQL code.  
Object oriented programming and object oriented mapping.  
Open source and free to use.

* **Cons**

The high level of abstraction can obscure what is actually happening in the implementation code

1. ***Explain the JPA strategy for handling Object Relational Mapping and important classes/annotations involved.***

I believe the Strategy is that a developer can work with object oriented programming and also work in object oriented mapping. By doing this, the developer avoid using JDBC / SQL code to map data.  
  
***Important annotations involved:***@Entity  
@Table  
@Column  
@Id  
@OneToOne  
@ManyToOne / @OneToMany  
@ManyToMany  
@JoinColumn

1. ***Outline some of the fundamental differences in Database handling using plain JDBC versus JPA***  
   In JDBC you work with the exact design of database while with ORM you work with java objects.  
   JDBC is using SQL language when mapping data.

JPA is a interface allowing you to use Hibernate, EclipseLink and many other languages which can exchange data between database and IDE. JPA supported languages are also called : **Java Persistence Query Language** (JPQL).

1. ***Explain some of the problems which occur when you write tests that involves database operations***

Forgetting to point the tests to the correct database.

Tests are depending on a database connection.

1. ***Explain ways (one strategy is enough) to mock away the database when writing unit tests.***

You could make a profile in the POM file so unit tests does not use a database.  
Have a persistence file only for unit tests which uses a embedded database?