Questions technique:

1. How to optimize sql query:

* Logic
* n'utiliser pas select \*
* WHERE is used instead of Having
* INNER JOIN is used instead of OUTER JOIN if possible
* User VIEW (for the security), Procedure
* Éviter utiliser DISTINCT
* Use EXIST, IN => normally use Exist
* INDEX
* JOIN au lieu de WHERE
* Optimise sub query, minimize the number of subquery

Interface : déclaration une collections de méthodes abstrait et static final variable (constant)

Use interface ou classe abstraite:

Classe abstrait :

1. You want to share code among several closely related classes.
2. You want to declare non-static or non-final fields
3. More detail, constructor

Interface:

1. Si on veut l'interface utilisé par plusieurs classes différentes
2. Si on veut profiter multi-héritage

Sérialisation :

La sérialisation: une façon d'écrire/transformer des objets en mémoire vers un flux de données binaires, ce procédé va donc nous permettre de rendre nos objets persistants.

SerialVersionUID: utilisé pour la désérialisation, vérifier que l’expéditeur and récepteur envoie et reçoit des objets de la même classe, si non => InvalidClassExceptions

* l'interface *Serializable*
* la classe *ObjectOutputStream : sérialiser et écrire : writeObject*
* la classe *ObjectInputStream : lire et déserialiser: readObject*

Une autre interface pour sérialiser: Externalizable

Extends Thread vs Implements Runable:

* Implementing Runnable should be preferred, flexible, can implements autre interface et extens une classe. Extend Thread is closed, stricted.
* Imp : can be use many time
* Imp: ne change pas, modifie pas le comportement de Thread
* Imp: les composant sont séparés, bien pour OOP. HAS au lieu ISA
* ExtendTh : classe hérite tous les methodes elle ne veut pas.

Index type:

1. B-tree: Indexe contenant des valeurs de clés dans une structure arborescente. <http://users.informatik.uni-halle.de/~brass/dbi13w/c4_index.pdf>.

Il effectue un parcours des différents blocs suivant un algorithme afin de trouver la feuille contenant le rowID pointant vers l'enregistrement souhaité.

1. - index Bitmap: index correspondant à un masque de bits pour les valeurs de colonnes indexées. Des AND et OR binaires permettent de faire des tests d'égalité.

Les index Bitmap sont adaptés à la manipulation de colonnes contenant des valeurs peu distinctes et aux requêtes contenant des prédicats d'égalité. Ils gèrent également mieux les valeurs NULL que les index B\*Tree.

Spring :

<bean id="petStore" class="org.springframework.samples.jpetstore.services.PetStoreServiceImpl">  
 <property name="accountDao" ref="accountDao"/>  
 <property name="itemDao" ref="itemDao"/>  
</bean>

ApplicationContext context = **new** ClassPathXmlApplicationContext("services.xml");  
PetStoreService service = context.getBean("petStore", PetStoreService.**class**);  
List<String> userList = service.getUsernameList();

* Maven
* Application.properties : datasource, view.prefix, view.suffix
* Application : SpringBootWebApplication (anotation @SpringBootWebApplication) avec main
* Entity Product
* ->interface ProductRepository extends CrudRepository<Product, Integer>
* ->interface ProductService
* -> ProductServiceImpl (anotation @Service et ProductRepository @Autowired)
* → ProductController (@Controller, ProductService @Autowired)
  + public ModelAndView showProduct(@PathVariable Integer id, Model model)
  + public ModelAndView saveProduct(Product product)
    - return new ModelAndView("redirect:/index");
  + public ModelAndView edit(@PathVariable Integer id, Model model)

@Autowired : cherche bean et l’injecter dans la classe ( dans la méthode setter, constructor)

J’ai fait pas mal des développement des bash shell pour lancer des cron job, le traitement des données, envoyer des emails aux utilisateurs, vérification du statut du serveur, redémarrer le serveur.

