```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;

import java.util.List;
import javax.persistence.EntityNotFoundException;
import javax.persistence.Query;
import metier.modele.Circuit;
import metier.modele.InfoPrincipale;
import metier.modele.Sejour;
import metier.modele.Voyage;

/**
 *
 * @author Administrateur
 */
public class VoyageDao {

    public static List<Voyage> findVoyageByNomPays(String nomPays) {

        Query query = JpaUtil.obtenirEntityManager().createQuery("select v "
                + "from Voyage v where v.paysDuVoyage.nom = :nom ");
        query.setParameter("nom", nomPays);
        List<Voyage> voyages = (List<Voyage>) query.getResultList();
        if (voyages == null) {
            throw new EntityNotFoundException("Impossible de trouver un voyage"
                    + " avec le nom : " + nomPays);
        }
        return voyages;
    }

    public static List<Voyage> listerVoyages() {

        Query query = JpaUtil.obtenirEntityManager().createQuery("select v "
                + "from Voyage v");

        List<Voyage> voyages = (List<Voyage>) query.getResultList();
        if (voyages == null) {
            throw new EntityNotFoundException("");
        }
        return voyages;
    }
    public static List<Circuit> listerCircuits() {

        Query query = JpaUtil.obtenirEntityManager().createQuery("select c "
                + "from Circuit c");

        List<Circuit> circuits = (List<Circuit>) query.getResultList();
        if (circuits == null) {
            throw new EntityNotFoundException("");
        }
        return circuits;
    }

    public static List<Sejour> listerSejours() {

        Query query = JpaUtil.obtenirEntityManager().createQuery("select s "
                + "from Sejour s");

        List<Sejour> sejours = (List<Sejour>) query.getResultList();
        if (sejours == null) {
            throw new EntityNotFoundException("");
        }
        return sejours;
    }

    public static List<Circuit> listerCircuitsParPays(String nomPays) {

        Query query = JpaUtil.obtenirEntityManager().createQuery("select c "
                + "from Voyage c where c in (select ci from Circuit ci ) "
                + " and c.paysDuVoyage.nom = :nom ");
        query.setParameter("nom", nomPays);
        List<Circuit> circuits = (List<Circuit>) query.getResultList();
        if (circuits == null) {
            throw new EntityNotFoundException("");
        }
        return circuits;
    }

    public static List<Sejour> listerSejoursParPays(String nomPays) {

        Query query = JpaUtil.obtenirEntityManager().createQuery("select s"
                + " from Voyage s where s in (select se from Sejour se) and "
                + "s.paysDuVoyage.nom = :nom ");
        query.setParameter("nom", nomPays);

        List<Sejour> sejours = (List<Sejour>) query.getResultList();
        if (sejours == null) {
```

```
        if (sejours == null) {
            throw new EntityNotFoundException("");
        }
        return sejours;
    }

    public static Voyage findVoyageById(Integer voyageId) {

        Voyage voyage = JpaUtil.obtenirEntityManager().find(Voyage.class,
                voyageId);
        if (voyage == null) {
            throw new EntityNotFoundException("Can't find voyage for ID "
                    + voyageId);
        }
        return voyage;
    }

    public static Voyage findVoyageByCodeVoyage(String codeVoyage) {

        Query query = JpaUtil.obtenirEntityManager().createQuery("select v "
                + "from Voyage v where v.CodeVoyage = :code ");
        query.setParameter("code", codeVoyage);
        Voyage voyage = (Voyage) query.getSingleResult();
        if (voyage == null) {
            throw new EntityNotFoundException("Can't find voyage for code "
                    + codeVoyage);
        }
        return voyage;
    }

    public static List<InfoPrincipale> listerInfos(String CV) {

        Query query = JpaUtil.obtenirEntityManager().createQuery("select "
                + "v.infos from Voyage v where v.CodeVoyage =:CV");
        query.setParameter("CV", CV);
        List<InfoPrincipale> infos = (List<InfoPrincipale>)
                query.getResultList();
        if (infos == null) {
            throw new EntityNotFoundException("pas d'infos pour ce voyage");
        }
        return infos;

    }

}
```