Bases de données II

TP2: Interfaçage de bases de données avec Hibernate

Plan

- Objectifs
- Étapes
- Directives

Plan

- Objectifs
- Étapes
- Directives

Objectifs du travail pratique

- Associations:
  - Mise en correspondance entre les tables d'une BD relationnelle et des classes Java
- Validation de données:
  - Se familiariser avec l'utilisation de *Hibernate Validator* pour valider les données au niveau applicatif
- Requêtes dans Hibernate:
  - Se familiariser avec l'utilisation de *Hiberante HQL* et *Query API* pour formuler des requêtes d'interrogation de la base de données
  - Réaliser des opérations CRUD sur les tables de la BD

Plan

- Objectifs
- Étapes
- Directives

Étapes

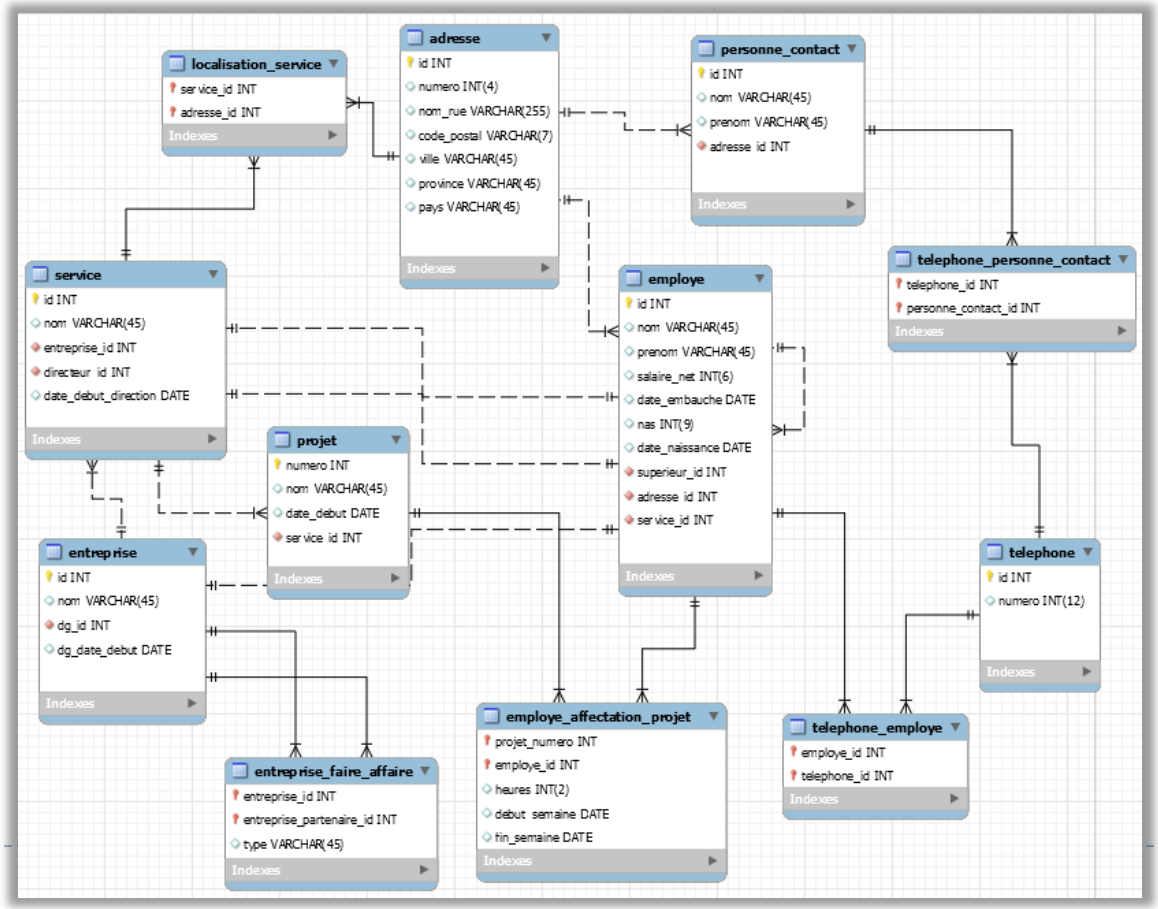
- Associations
- Validation de données
- Requêtes

Étapes

- Associations
- Validation de données
- Requêtes

Associations : Modèle relationnel de la BD

- Le schéma de la BD à considérer est celui illustré par l'image ci-dessous
- Remarque: Le fichier source de ce modèle, réalisé avec MySQLWorkbench, est disponible sur le site du cours



Associations : énoncé

1. Générez manuellement les entités Java qui représentent les classes du domaine telles que décrites par les différentes tables du schéma relationnel précédent
2. Expliquez et justifiez la raison d'être et la pertinence de:
   - Chaque entité créée
   - De chaque association (1-1, 1-N, N-M)

- Remarques:
  - Les explications et justifications sont à intégrer aussi bien au niveau des classes Java que dans le rapport écrit qui accompagne le code
  - Utiliser des annotations et non du XML pour la configuration des entités

Étapes

- Associations
- Validation de données
- Requêtes

Hibernate Validator: introduction

- Hibernate Validator est le package Java offert par Hibernate pour développer des stratégies de validation des données au niveau des entités Hibernate

- La validation des données permet d'ajouter des contraintes pour assurer la cohérence et l'intégrité des données vis-à-vis de la logique métier du domaine considéré

- C'est une implémentation indépendante de Hibernate ORM

Hibernate Validator: énoncé

1. Téléchargez et intégrez dans votre projet Java le package Hibernate Validator
   - http://hibernate.org/validator/
2. Explorez et documentez l'utilisation des contraintes suivantes:
   1. @NotEmpty, @Valid, @NotBlank, @Pattern, @ConstraintComposition,@Size, @Min, @Max, @Email, @URL, @Future, @Past, @NULL, @Digits, @DecimalMin, @DecimalMax, @Length, @Range
3. Créez un validateur personnalisé:
   1. Il permettra de s'assurer qu'un code postal n'est pas NULL et qu'il est de la forme: LCL CLC (L: lettre, C: chiffre)
   2. L'annotation qui sera créée sera nommée: @CodePostalValide
   3. Le processus de création est à documenter
4. Appliquez le plus grand nombre de ces validateurs à vos entités
   1. Vous avez la liberté de suggérer des contraintes réalistes sur les différents modèles

Étapes

- Associations
- Validation de données
- Requêtes

Hibernate HQL et Query API: introduction

- HQL est similaire à SQL
  - Utiliser des classes à la place des tables
  - Utiliser des attributs à la place des colonnes
- HQL est indépendant de la base de données utilisée
  - Le code SQL propre à chaque SGBDR est généré via l'adaptateur approprié

Hibernate HQL et Query API: énoncé

1. Explorez et documentez l'utilisation de:
    1. La classe *Query* pour formuer des requêtes HQL
    2. Les clauses suivantes:
        1. from, as, select, where, order by, group by, update, delete, les fonctions d'agrégation et les sous requêtes
    3. Paramètres nommés
    4. La pagination
    5. Itération sur les résultats retournés
    6. Requêtes nommées (@NamedQuery)
    7. Différents types de jointures
2. Appliquez les différents éléments du point 1 sur le schéma de la base de données du TP. Il est recommandé d'intégrer ce point dans la réalisation du point 1
3. Fournissez les méthodes qui implémentent les différentes opérations CRUD sur les tables de la BD:
    1. Du code de Java pour tester les différentes méthodes devra être fournit

Plan

- Objectifs
- Étapes
- Directives

Directives

- Le travail est à réaliser en équipes de deux
- Ce que vous avez à remettre:
  - Code source documenté
  - Un rapport couvrant les trois parties du TP ainsi que:
    - Explications des choix et décisions prises lors de la réalisation de ce TP
    - Références utilisées
    - Les noms des coéquipiers sont à indiquer dans la page de garde du rapport ainsi que dans chaque classe de votre projet
- Chaque groupe aura à remettre un seul fichier compressé comprenant le code source + le rapport
- La remise se fera directement sur le site du cours
- Dernier délai pour la remise de cette partie du projet:
  - Lundi 7 décembre 8h30 (pas de possibilité d'extension de délai)

Évaluation

- Lors de l'évaluation, les éléments suivants seront pris en considération:
  - Aptitude à aller chercher les éléments manquants et la validation de tout choix/décision/interprétation
  - Complétude
  - Respect des consignes
  - Qualité du rapport
  - Une présentation/rencontre est exigée. Cette séance aura lieu le 07 décembre 2015 au J-432 à partir de 16h05

Bonus!

- +15% de votre note du TP (note + bonus ne va pas dépasser 100% … avec possibilité de reporter le surplus de points sur le TP01!) pour la couverture du package *Hibernate Search (indexing and searching full-text with Hibernate)*:
  - http://hibernate.org/search/