Implémentation des SGBDs L2

Projet

Projet de gestion ferroviaire

Introduction

L'objectif de ce projet est de réaliser un gestionnaire pour les trajets des trains sur un réseau ferroviaire. On souhaite que la gestion puisse être effectuée par plusieurs personnes à la fois. Ce gestionnaire prendra la forme d'une application Web développée en Java utilisant une base de données Postgresql. Une attention particulière sera portée à la gestion de la concurrence, au maintient de la cohérence des données et plus généralement à l'intégration du SGDB dans l'application.

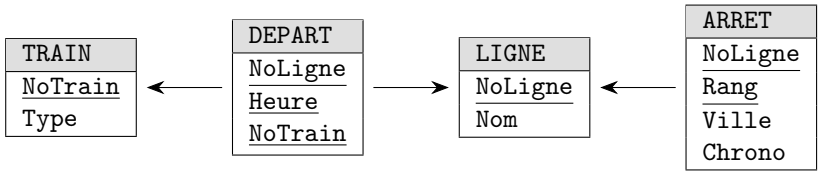
Vocabulaire

Un parcours sur le réseau ferroviaire est appelé une *ligne*, elle est identifiée par un *numéro* et porte un nom comme par exemple "Clermont-Paris". La gare de départ et le terminus d'une ligne font partis de l'ensemble de ses *arrêts*. Les gares sont assimilées aux *villes* pour simplifier. Les arrêts d'une ligne sont ordonnés par *rang* croissant avec zéro pour le départ : le rang correspond à l'ordre de visite des arrêt. Le temps écoulé entre le départ d'un train au début d'une ligne et son arrivée dans une gare est appelé le *chrono*. Dans la table `DEPART` se trouve les *horaires* d'arrivée de chaque train à la première gare d'une ligne. Une même ligne peut être empruntée par plusieurs trains dans une même journée. Pour simplifier, on supposera que **les horaires des trains sont les mêmes tous les jours**. Un train est identifié par son *numéro* et possède un *type* qui est soit TGV, TER ou INTERCITE.

Schéma de la base de données

Le schéma initial de la base de données est le suivant.

L'heure de départ `Heure` et `Chrono` sont exprimés en heure. Ils sont stockés sous la forme d'un nombre décimal. Pour simplifier leur lecture, ces valeurs sont représentées avec la notation classique des heures. Ainsi, "9h30" est stockée sous la forme du nombre 9,5.



Ci-dessous, un exemple de contenu pour les différentes tables de la base de données.

TRAIN	
NoTrain	Type
9634	INTERCITE
3472	TER
4486	TGV
8645	TGV

DEPART		
NoLigne	Heure	NoTrain
2	6h00	4486
0	9h00	9634
1	9h00	3472
2	9h30	8645
3	15h00	9634

LIGNE	
NoLigne	Nom
0	Clermont-Paris
1	Clermont-Lyon
2	Marseille-Paris
3	Paris-Clermont

ARRET			
NoLigne	Rang	Ville	Chrono
0	0	Clermont-Fd	0
0	1	Vichy	1h
0	2	Paris	4h
1	0	Clermont-Fd	0
1	1	Lyon	1h
2	0	Marseille	0
2	1	Lyon	1h
2	2	Paris	3h
3	0	Paris	0
3	1	Clermont-Fd	4h

Cahier des charges

La section suivante propose d'implémenter certains objectifs. Ces objectifs doivent être réalisés en gardant en tête le cahier des charges suivant :

Éviter les erreurs HTTP 500

Éviter dès que cela est possible, les erreurs HTTP 500. Pour cela, il faut rattraper au maximum les exceptions soulevées par les erreurs de violation de contraintes au niveau de la base de données, les erreurs dues à des données incorrectes dans les requêtes HTTP POST ou GET, etc. Vous devez dans ces cas, retourner une erreur HTTP 400 avec un message expliquant précisément la cause de l'erreur. Ne pas oublier de retourner une erreur 404 pour les résultats vides.

Choisir le bon niveau d'isolation

Pour chaque transaction, il faut choisir le niveau d'isolation qui permet de s'assurer qu'aucune ou très peu d'anomalie de sérialisabilité ne puisse apparaître, tout en maximisant la concurrence entre les transactions. Justifier votre choix dans un commentaire au début de chaque transaction.

Recommencer les transactions avec une erreur de sériabilité

En cas d'exceptions soulevées par une erreur dues à un interblocage ou un conflit transactionnel, recommencer la transaction en cours sans retourner d'erreur.

Implémenter au maximum les contraintes simples dans le SGBD

Implémenter systématiquement les contraintes sur les données avec des clés primaires, clés étrangères ou **CHECK** dès que cela est possible.

Objectifs

Dans cette section se trouve les objectifs obligatoires du projet.

Objectif 1. Compléter la classe "com.uca.dao.DBInitializer" de sorte que chaque table du schéma soit crée si elle est manquante dans la base de données.

Objectif 2. En vous inspirant des pages existantes pour les trains, créer les pages suivantes pour gérer les lignes :

— une page qui liste l'ensemble des lignes avec leur numéro et leur nom à l'URL `http://localhost:8081/ligne` avec des boutons pour supprimer chaque ligne,

— une page avec un formulaire pour créer une nouvelle ligne à l'URL : `http://localhost:8081/ligne/ajout`.

Objectif 3. De même, créer les pages suivantes pour gérer les arrêts :

— une page affichant dans l'ordre les arrêt d'une ligne de numéro n à l'URL `http://localhost:8081/arret?noligne=n` avec des boutons pour supprimer chaque arrêt,

— une page avec un formulaire pour ajouter un arrêt sur une ligne n au rang k à l'URL `http://localhost:8081/arret?noligne=n&rang=k`.

Ajouter les liens suivants :

— sur la page affichant les lignes, un lien vers la page affichant les arrêts de cette ligne,

— sur la page affichant les arrêts d'une ligne, un lien pour ajouter un arrêt à la fin de la ligne.

Objectif 4. Finalement, créer les pages suivantes pour gérer les départs :

— pour chaque train de numéro n, une page listant les départs de ce train à l'URL `http://localhost:8081/depart?notrain=n` avec des boutons pour supprimer chaque départ,

— une page de formulaire pour ajouter un départ à un train à l'URL `http://localhost:8081/depart/ajout?notrain=n`.

Ajouter les liens suivants sur la page affichant les trains :

— un lien vers le formulaire pour ajouter un départ à chaque train,

— un lien vers la page affichant les départs de chaque train.

Objectif 5. Lors de la suppression d'un train, supprimer tous ses départs automatiquement. Lors de la suppression d'une ligne, de même supprimez tous ses arrêts et les départs sur cette ligne.

Objectif 6. Lors de la suppression d'un arrêt, veuillez à ce que tous les arrêts suivants, c'est à dire de rang strictement supérieur soient décalés d'un rang en arrière. Lors de l'ajout d'un arrêt à un rang déjà existant, décaler les arrêts de rang supérieur d'un rang en avant, avant d'insérer le nouvel arrêt. De plus, quelque soit les requêtes du client, on souhaite que sur chaque ligne avec n arrêts, le rang de ses arrêts soit entre 0 et $n - 1$.

Sur la page affichant les arrêts d'une ligne, ajouter des liens vers le formulaire d'ajout d'un arrêt pour pouvoir ajouter un nouvel arrêt avant chaque arrêt existant.

Objectif 7. Pour des raisons de sécurité, on souhaite que sur chaque ligne le temps de passage d'un arrêt au suivant, c'est à dire que la différence de leur chronos, soit supérieur ou égale à 15 minutes. Assurer que cette contrainte est toujours vérifiée.

Objectif 8. Pour des raisons de cohérence, vérifier qu'à tout instant que chaque train n'est pas à deux (ou plus) endroits à la fois.

Objectif 9. On suppose qu'un train qui arrive au bout d'une ligne reste à la gare terminus sur une voie de réserve. Il ne peut que repartir sur une ligne dont la gare de départ est le terminus de la ligne qu'il vient de terminer. Vérifier aussi que ça soit le cas d'un jours à un autre. Dit autrement, les trains ne se téléportent pas et se déplacent uniquement en empruntant des lignes. (cette propriété n'est pas respectée par l'exemple ci-dessus).

Objectifs bonus

Objectif 10. Ajouter un moyen de faire des recherches d'itinéraire, d'abord sans correspondance puis avec une correspondance. On s'intéressera uniquement aux itinéraires effectués dans une même journée.

Objectif 11. Améliorer le style du site Web et sa simplicité d'utilisation. Par exemple, faire en sorte que les champs des formulaires dont les valeurs possibles sont connues (par ex. celle du type d'un train, d'une gare pour la recherche d'itinéraire) soient représentés par une balise HTML `<select>`.

Objectif 12. En modifiant le schéma de la base, ajouter un nombre de voie de réserve à chaque gare et vérifier que le nombre de train en réserve (c'est à dire attendant à une gare pour partir sur une ligne) n'excède jamais le nombre de voie de réserve.

Objectif 13. En modifiant le schéma de la base, ajouter une longitude et latitude à chaque gare. Sur la page d'accueil du site, ajouter une carte avec la position de chaque gare, le trajet de chaque ligne et la position à l'heure actuelle de chaque train.