Emile Ndagijimana *M1 MIAGE*

Thomas Lienard

Joris Henry

Boris Thouvenin

Rapport Projet ABD

29 Janvier 2015

I. Introduction

Ce rapport à pour but de vous présenter le projet que nous avons réalisé. Pour rappel ce projet consistait à établir trois application Java pour la gestion d'un parc de location de vélos sur la ville de Nantes via le protocole JDBC et une base de données sur un serveur distant. Les trois applications sont respectivement pour les clients (Location, réservation, abonnement), les conducteurs de véhicules de régulations pour leurs routines et donner des feedbacks, et la dernière pour les superviseurs, afin qu'ils puissent définir les routines et les ordres. Chacune de ces personnes pourra accéder aux applications avec des droits d'accès différents.
Nous ferons tout d'abord un gros retour sur les rapports intermédiaires que nous avons rendu et nous verrons notamment les changements correctifs que nous avons appliqué suite à vos retours et à l'avancé du programme. Une fois cette partie passée nous regarderons plus profondement le programme que nous vous avons rendu, le décrirons et le commenterons à l'aide de quelques scénarios qui mettront en évidence

II. Retour sur les Rapports intermédiaires

Nous allons donc commencer par faire un retour sur les différents rapports intermédiaires. Ceux-ci étant loins d'être parfait dû à la dfficulté de bien cerner le projet et à poser nos pensées pas écrit. Nous avons donc au fil du projet et avec vos remarques, repensé le schéma conceptuel de données ainsi que le schéma relationnel complet pour en arriver à la version finale :

Schéma relationnel avec les contraintes de domaines

Clients(IdC, CB, code)

- dom (IdC) : entier positif non nul
- dom (CB), dom(code) : entier positif de 16 et 8 chiffres non nulle

ClientAbo(#IdC, nom, prénom, dateNais, adr, remise, dateAbo)

- ClientAbo (IdC) ⊂ Clients (IdC)
- dom (nom), dom (prenom), dom (adr), : Chaine de caractères non nulle
- dom (dateNais), dom(dateAbo) : dates, granularité (YYYY-MM-DD)
- dom (remise) : "Booléen" {true,false}

ClientNonAbo(#IdC)

- ClientAbo (IdC) ⊂ Clients (IdC)

Velos (IdVelo, modele, dateMS, etat)

- dom (idVelo) : entier positif non nul
- dom (modele), dom (etat) : chaine de caractères non nulle (pour etat : {Ok,HS})
- dom (dateMS) : date, granularité (YYYY-MM-DD)

Stations(IdS,adresse,capMax,nbVelo)

- dom (IdS), dom (capMax), dom (nbVelo) : entier positif non nul
- dom (adresse) : chaine de caractères non nulle

Valeur (IdValeur, #IdS, etat (Vplus,Vmoins), dateDeb,dateFin)

- Valeur (IdS) c Stations (IdS)
- dom (IdValeur), dom (IdS) : entier positif non nul
- dom (dateDeb), dom (dateFin) : date, granularité (HH24:MI:SS)

Bornettes(IdB, #IdS, #IdVelo, etat)

- Bornettes (IdS) c Stations (IdS), Bornettes (IdVelo) c Velos (IdVelo)
- dom (IdB), dom(IdS) : entiers positifs non nul
- dom(IdVelo) : entiers positifs
- dom (etat) : chaine de caractère non nulle {Libre,Occupe,HS}

Vehicules(IdVehicule, modele, dateMx, capacite)

- dom (IdVehicule), capacite : entiers positifs non nul
- dom (modele) : chaine de caractere non nulle
- dome (dateMX) : date, granularité : (YYYY-MM-DD)

Ordre(IdO, defi, numOrd)

- dom (IdO), dom (numOrd) : entier positif non nul
- dom (defi) : chaine de caractère non nulle

Routines(IdR, #IdVehicule, dateR)

- Routines (IdVehicule) c Vehicules(IdVehicule)
- dom (IdVehicule), dom (IdR) : entier positif non nul
- dom (dateR) : date, granularité (YYYY-MM-DD)

Remise(IdR, code, dateDeb, dateFin)

- dom (IdR), dom (code) : entier positif non nul
- dom (dateDeb), dom (dateFin) : date, granularité (YYYY-MM-DD)

Amendes(IdA,#IdC, dateAm, statut)

- Amendes (IdC) c Clients (IdC)
- dom (IdA), dom (IdC) : entier positif non nul
- dom (dateAm) : date, granularité (YYYY-MM-DD HH24:MI:SS)
- dom (statut) : chaine de caractère non nulle {Paye,Impaye}

- Reservations (IdR)
- dom (IdR) : entier positif non nul

ResaJour (#IdR, dateRes)
- ResaJour (IdR) c Reservations (IdR)
- dom (dateRes) : date, granularité (YYYY-MM-DD)

ResaJourRepete (#IdR, dateJour, dateDeb, dateFin)
- ResaJourRepete (IdR) c Reservations (IdR)
- dom (dateJour), dom (dateDEb), dom (dateFin) : date, granularité (YYYY-MM-DD)

ResaPeriode (#IdR, dateDeb, dateFin)
- ResaPeriode (IdR) c Reservations (IdR)
- dom (dateDeb), dom (dateFin) : date, granularité (YYYY-MM-DD)

Association:

louer(IdC, IdVelo, dateLoc)
- louer (IdC) c Clients (IdC), louer (IdVelo) c Velos (IdVelo)
- dom (dateLoc) : date, granularité (YYYY-MM-DD HH24:MI:SS)

deplacer(IdVehicule, IdVelo)
- deplacer (IdVehicule) c Vehicules (IdVehicule), deplacer (IdVelo) c Velos (IdVelo)

accueil(IdVelo, IdB)
- accueil (IdVelo) c Velos (IdVelo), accueil (IdB) c Bornettes (IdB)

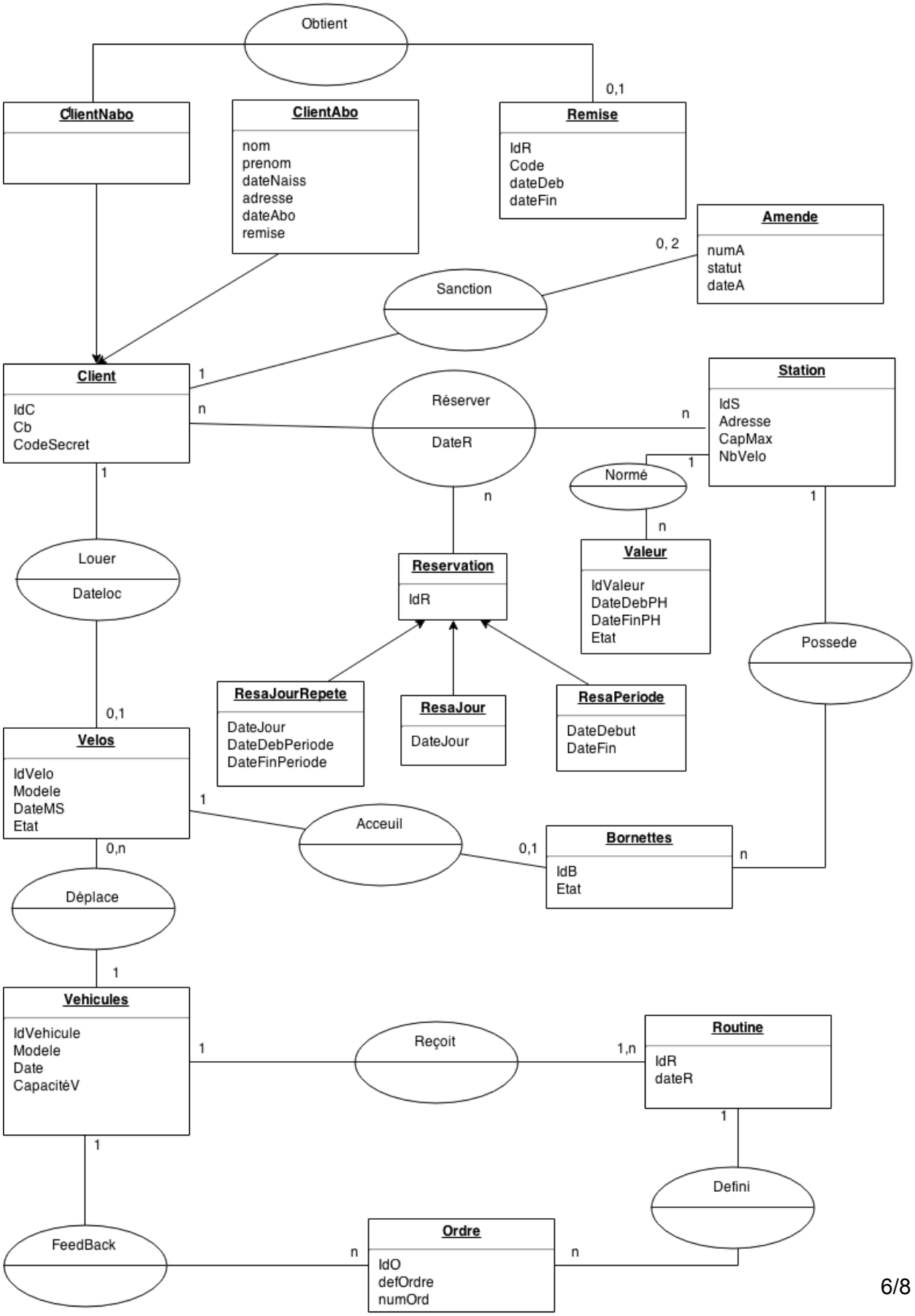
defini(IdR,IdO)
- defini (IdR) c Routines (IdR), defini (IdO) c Ordres (IdO)

possede(IdS, IdB)
- possede (IdS) c Stations (IdS), possede (IdB) c Bornettes (IdB)

reserver(IdR, IdC, IdS, dateR)
- reserver (IdR) c reservations (IdR), reserver (IdC) c clientAbo(IdC), reserver (IdS) c stations (IdS)
- dome (dateR) : date, granularité (YYYY-MM-DD)

feedback(IdO,IdVehicule,etat,notification)
- feedback (IdO) c Ordres (IdO), feedback (IdVehicule) c Vehicules (IdVehicule)
- dom (etat) : chaine de caractere non nulle {Valide,Annule}
- dom (notification) : chaine de caractere

Schéma conceptuel de données:


Obtient
0,1
ClientNabo
ClientAbo
nom
prenom
dateNaiss
adresse
dateAbo
remise
Remise
IdR
Code
dateDeb
dateFin
Amende
numA
statut
dateA
0, 2
Sanction
Client
IdC
Cb
CodeSecret
1
n
Réserver
DateR
n
Station
IdS
Adresse
CapMax
NbVelo
1
Normé
n
1
Louer
Dateloc
Reservation
IdR
Valeur
IdValeur
DateDebPH
DateFinPH
Etat
Possede
ResaJourRepete
DateJour
DateDebPeriode
DateFinPeriode
ResaJour
DateJour
ResaPeriode
DateDebut
DateFin
0,1
Velos
IdVelo
Modele
DateMS
Etat
Acceuil
0,1
Bornettes
IdB
Etat
n
0,n
Déplace
1
Vehicules
IdVehicule
Modele
Date
CapacitéV
Reçoit
1,n
Routine
IdR
dateR
Defini
FeedBack
n
Ordre
IdO
defOrdre
numOrd
n

Choix de conception :

- Routines

Une routine est composée d'ordre. Chaque ordre est identique mais peut appartenir à deux routines différentes. Ainsi lorsqu'un superviseur donne le même ordre sur deux routines différentes parce qu'il le veut au plus tôt, la première routine qui a validé l'ordre avant l'ordre "au plus tôt" déclenche un trigger qui va annuler l'ordre sur les autres routines.

- Alertes

On ne prend pas en compte les alertes, on ne laisse pas le choix au véhicule dans l'application : soit il valide, soir il notifie

- Paiement

Un client paye au moment où il loue le vélo, la remise est calculée à ce moment là.

- Location

Un client abonné ne peut pas louer un vélo si il a une réservation pour la même période dans une autre station.

- Ordres

Les ordres ne sont pas standardisés

III. Description des scénarios

Afin de tester nos triggers, voici les différents scénarios que nous devront mettre en place :

- Réservations simultanées de deux clients dans une même station pour une même période alors qu'il ne reste qu'un seul vélo disponible dans cette station à cet horaire la.
  Cela permet de montrer l'utilité du niveau d'isolation sérializable ainsi que différents triggers comme la vérification de la durée d'abonnement, de la disponibilité en vélo et si le client à déjà une réservation à cet horaire.

- Différentes locations de vélos pour tester les conditions de location (amendes, état du vélo, réservation concurente, location concurente).

- Une exécution de routine par un véhicule

Conclusion

Ce projet fût plutôt facile à prendre en main, les rapports intermédiaire permettent de vraiment sentir les contraintes qu'ils va falloir insérer dans la base et vérifier dans l'application.
Nous étions dans les temps durant toute la phase de conception mais malheureusement nous nous avons commencer à coder les requêtes trop tard et rencontré trop de problèmes pour pouvoir terminer dans les temps.
Un week-end de plus aurait été necessaire pour développer des jeux de test vraiment complets.

Malgré tout, nous sommes désormais tous beaucoup plus à l'aise avec les concepts de triggers, de vérouillage et d'isolation car nous avons tous touché à tous les aspects du projet. Le but est donc atteint de ce coté là.