CETE Méditerranée

Université Aix-Marseille
IUT Aix en Provence

REQUÊTES SUR BASES DE DONNÉES DE TRANSPORT COLLECTIF

Rapport de stage

NOTICE ANALYTIQUE

Organisme commanditaire : CETE Méditerranée, DGITM, CERTU		
Titre : Requêtes sur Bases de Données de Transports Collectif		
Sous-titre : Cartes thématiques des arrêts et lignes de TC	**Langue** : Français	
Organismes auteurs **Université AMU, CETE Med**	**Rédacteurs ou coordonnateurs** Lucas JOLLY, Patrick GENDRE	**Date** Juin 2013

Résumé :

Le CETE Méditerranée contribue à un travail lancé par le CERTU visant à constituer progressivement une Base de Données (BD) nationale sur les Transports Collectifs (TC), en vue de répondre aux besoins d'études des services (services centraux tels que le Service d'Observation Statistique des Transports, SETRA ou CERTU), mais aussi des services déconcentrés régionaux et départementaux, et CETE. Les données sont détenues par les collectivités locales (autorités organisatrices de transport, AOT) auxquelles il faut les demander, ce qui prendra plusieurs mois.

Dans un premier temps, il s'agit de commencer à travailler avec les données disponibles (les données open data par exemple celles de Rennes), puis progressivement avec celles que les collectivités AOT mettront à disposition du CERTU dans le cadre de conventions d'utilisation.

La base de données serait à terme rendue disponible aux chargés d'études transport du ministère sous forme de couches utilisables dans un Système d'Information Géographique (SIG) afin de produire des cartes et analyses territoriales.

Deux principaux types de données seront publiées sous forme de couches : les arrêts et les tronçons de lignes, complétés chacun par des indicateurs permettant d'en tirer des informations utiles sur l'offre de transport public.

Ce document présente le travail effectué dans le cadre du stage de Lucas Jolly, étudiant en 2ème année à l'IUT d'Aix-en-Provence, dont l'objectif est de mettre au point une première version de la chaîne technique (en particulier les requêtes SQL de calcul des indicateurs) produisant les couches SIG attendues par le CERTU avec des données réelles (celles qui ont été publiées en licence open data, et notamment les données du réseau Star de Rennes). Les requêtes de calcul d'indicateurs sont faits dans la base de données de l'application Chouette qui permet d'importer les données.

Les couches suivantes sont produites :
- arrêts physiques (une autre couche pourrait être créée avec les arrêts commerciaux) : nombre de passages à une date donnée, amplitude horaire pour la même date, nombre d'habitants résidant à moins de 300mètres
- tronçons de ligne : vitesse moyenne approximative à vol d'oiseau entre arrêts successifs.

Les fichiers issus de données open data seront rendus disponibles en téléchargement.

Mots clés : transport public, accessibilité transport, logiciel libre, base de données, points d'arrêt		**Diffusion :** Version électronique	
Nombre de pages : 32 pages		**Confidentialité :** Non	**Bibliographie** : Oui

Mise à jour du 21/06/13

Table des matières

INTRODUCTION 6

CONTEXTE 6
LE CETE MÉDITERRANÉE 6
VERS UNE BD NATIONALE DE L'OFFRE DE TRANSPORT COLLECTIF POUR LES ÉTUDES DU MINISTÈRE 7
REQUÊTES SPATIALES ET SIG TRANSPORT OPEN SOURCE 7
OBJECTIFS DU STAGE 7
DÉMARCHE 9
CONTENU DU RAPPORT 9

ENVIRONNEMENT TECHNIQUE 9

LES DONNÉES 9
LES DONNÉES DE TRANSPORT PUBLIC 10
LES DONNÉES DÉMOGRAPHIQUES 10
LES LOGICIELS 11
CHOUETTE ET SES COMPOSANTS 11
QGIS 12
POSTGRESQL ET SES COMPOSANTS 12

MISE EN PLACE DE L'ENVIRONNEMENT DE TRAVAIL 13

INSTALLATION DE L'ENVIRONNEMENT DE TRAVAIL 13
CRÉATION DE LA BASE DE DONNÉES ET MISE EN PLACE DE L'APPLICATION 13
POSTGRESQL 13
CHOUETTE 14
Installation 14
Installation (variante) 14
Utilisation 14
POSTGIS 15
QGIS 15
DONNÉES DÉMOGRAPHIQUES 16

CONCEPTION DES REQUÊTES 16

LES INDICATEURS À CALCULER 16
PRÉSENTATION DE LA BASE ET DU MODÈLE MÉTIER TC 18
TRAVAIL INITIAL : PROJET POTIMART 20
INDICATEURS DES POINTS D'ARRÊT 21
Description 21
Méthode de calcul 21
NOMBRE DE PASSAGES UN JOUR DONNÉ 22
Description 22
Méthode de calcul 22
NOMBRE D'HABITANTS À PROXIMITÉ 24
Description 24
Méthode de calcul 24
INDICATEURS DES TRONÇONS DE LIGNES 25
VITESSE MOYENNE 25
Description 25

Méthode de calcul....25
NOMBRE D'HABITANTS MOYEN PAR ARRÊT POUR UNE LIGNE....27
Description....27
Méthode de calcul....27
TEST AVEC LES DONNÉES OPEN DATA DE RENNES....28
RÉUTILISATION DES RÉSULTATS....29

CONCLUSIONS ET SUITES À DONNER....30

EN RÉSUMÉ....30
PERSPECTIVES....31

ANNEXE : RÉFÉRENCES....32

Remerciements.

Nous remercions Michel Etienne, Bernard Rongione et Charles Rapenne.

I. INTRODUCTION

Le présent rapport décrit le travail effectué par Lucas Jolly lors de son stage effectué au CETE Méditerranée d'avril à juin 2013.

A. Contexte

1. Le CETE Méditerranée

Le CETE (Centre d'Etudes Techniques de l'Equipement) Méditerranée est un service technique du ministère du développement durable. http://www.cete-mediterranee.fr.
Avec environ 400 agents, ses activités sont très nombreuses : études, expertises, conseils, assistance à maîtrise d'œuvre, recherche, méthodologie, animations de réseaux, formation, avis technique, essais de laboratoire et contrôles de chantier. Le CETE Méditerranée fait partie du réseau scientifique et technique du Ministère qui deviendra un établissement public début 2014, baptisé CEREMA[1].

Le CETE Méditerranée contribue à l'action du ministère du développement durable en vue de développer l'information sur tous les modes de transports, dite 'information multimodale'. Mon stage a été effectué au sein du département DCEDI, dans la mission MIM. La mission Information Multimodale[2] est un petit service de 2 personnes qui contribue à l'action de l'Agence Française de l'Information Multimodale et de la Billettique[3], au travail de capitalisation des connaissances et d'animation technique du CERTU (un service technique central du ministère basé à Lyon), ainsi qu'au programme de recherche et développement pour l'innovation dans les services de mobilité PREDIM (www.predim.org). Dans ce cadre, la mission contribue en particulier à la mise en œuvre d'un annuaire des sources d'information transport www.passim.info et d'un logiciel libre www.chouette.mobi permettant la validation, l'échange et l'édition de données décrivant l'offre de réseaux de transport public conformément à un profil d'échange normalisé (XML Neptune).

1 http://www.cerema.fr

2 http://www.cete-mediterranee.fr/tt13/www/rubrique.php3?id_rubrique=27

3 http://www.cete-mediterranee.fr/tt13/www/rubrique.php3?id_rubrique=27

2. Vers une BD nationale de l'offre de transport collectif pour les études du ministère

Le CETE Méditerranée contribue à un travail lancé par le CERTU en vue de la constitution progressive d'une Base de Données (BD) nationale sur les Transports Collectifs (TC), en vue de répondre aux besoins d'études des services (services centraux tels que le Service d'Observation Statistique des Transports, SETRA ou CERTU), mais aussi des services déconcentrés régionaux et départementaux, et CETE. Les données sont détenues par les collectivités locales (autorités organisatrices de transport, AOT) auxquelles il faut les demander, ce qui prendra plusieurs mois. Dans un premier temps, il s'agit de commencer à travailler avec les données disponibles (les données open data par exemple celles de Rennes qui sont de bonne qualité), puis progressivement avec celles que les collectivités AOT mettront à disposition du CERTU dans le cadre de conventions d'utilisation. La base de données sera rendue disponible aux chargés d'études transport du ministère sous forme de *couches* utilisables dans un Système d'Information Géographique (SIG) afin de produire des cartes et analyses territoriales. Deux principaux types de données seront publiées sous forme de couches : les arrêts et les tronçons de lignes, complétés chacun par des indicateurs permettant d'en tirer des informations utiles sur l'offre de transport public.

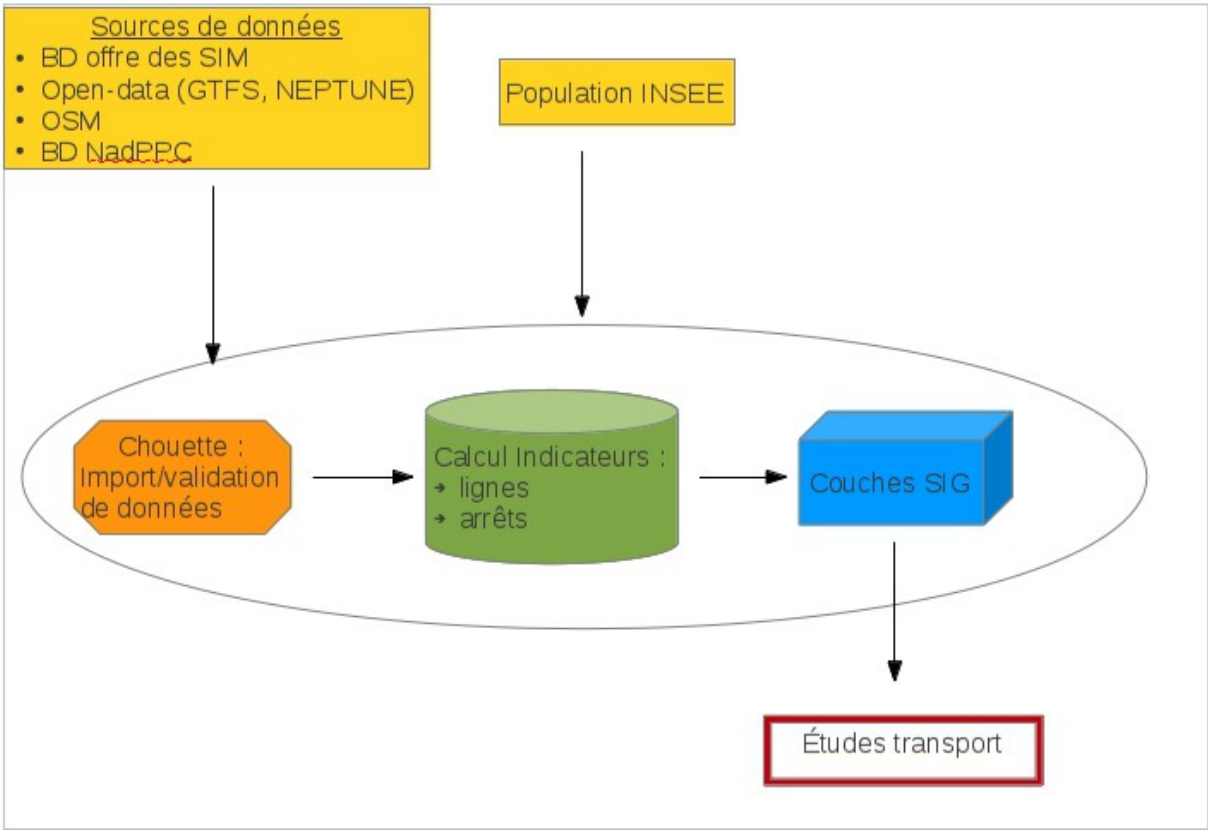
3. Requêtes spatiales et SIG transport *open source*

Le CETE Méditerranée a participé en 2007-2008 à un projet visant à développer des solutions SIG transport open source pour l'information multimodale (Potimart).
Dans le prolongement du projet Potimart, fin 2008, le CETE a confié à la société MOBIGIS le développement d'un démonstrateur d'une base de données permettant de produire des analyses simples sur des données de voirie routière et de transport collectif. La 1ère phase de cette étude avait permis de développer des requêtes ou scripts SQL permettant de produire notamment des indicateurs relatifs à la fréquence et vitesse des lignes d'un réseau TC.
Ces scripts ont été publiés sur le web (https://github.com/Potimart/Potimart et http://www.cete-aix.fr/tt13/www/article.php3?id_article=192).

B. Objectifs du stage

L'objectif de ce stage est de contribuer au travail du CERTU de constitution de couches de données sur les arrêts et lignes de Transport Collectif produites sur l'ensemble du territoire national et diffusables aux services d'études du ministère.
Il s'agit de mettre au point une première version de la chaîne technique, en particulier les requêtes SQL de calcul des indicateurs, produisant les couches SIG attendues par le CERTU avec des données réelles sur des territoires français.

Le schéma ci-dessous donne une vue d'ensemble de la chaîne de traitement qu'il est envisagé de mettre en place pour ce projet :



La partie entourée représente la zone d'action durant le stage.

Les données de TC peuvent provenir de différentes sources et être disponibles à différents formats (en priorité celles qui seront mises à disposition par les AOT dans le cadre de convention avec le ministère, mais en attendant leur disponibilité complète, également de données open data publiées par les collectivités ou les transporteurs, ainsi que des données OpenStreetMap éventuellement).

Dans le cadre de ce stage, nous travaillerons uniquement avec des données open data existantes, en s'appuyant pour cela sur le logiciel libre Chouette, qui permet d'importer / exporter / gérer des données de transport collectif.
Pour calculer les indicateurs proposés par le CERTU, nous compléterons les données par des données de population de l'Insee.

Notre travail consiste à développer la version initiale des scripts ou requêtes SQL permettant de calculer les indicateurs, ainsi qu'à mettre en place l'environnement permettant, à partir des fichiers de données initiaux, de produire les fichiers SIG utilisables pour afficher les arrêts et tronçons de ligne de TC.
Pour cela, nous partirons de requêtes SQL, permettant le calcul d'indicateurs relatifs aux réseaux de TC, écrites dans le cadre du projet POTIMART, que nous adapterons.
De plus, il nous faudra écrire nos propres scripts SQL pour créer de nouvelles tables d'arrêts ou de lignes à partir de la BD Chouette ou pour obtenir de nouveaux indicateurs notamment en liaison avec des données démographiques de l'INSEE.

C. Démarche

Pour ce stage, les différentes étapes du travail effectué ont été les suivantes :

- ➔ Prise de connaissance du contexte
- ➔ Installation de l'environnement de travail
- ➔ Test avec des jeux de données
- ➔ Écriture de script SQL
- ➔ Perspectives d'utilisation
- ➔ Analyse des résultats obtenus
- ➔ Test de la portabilité
- ➔ Bilan et suites à donner.
- ➔ Mémoire de fin de stage.

D. Contenu du rapport

Outre la présente introduction, ce rapport téléchargeable sur le site web du CETE comprend 3 parties :

- Contexte technique décrivant les données et logiciels utilisés .
- Mise en place de l'environnement de travail détaillant la procédure à suivre pour obtenir les outils de réalisation du projet .
- Conception des requêtes détaillant le traitement effectué sur la base.

Il se termine par une conclusion proposant des pistes pour continuer, et en annexe, par une liste des principales références utiles.

II. ENVIRONNEMENT TECHNIQUE

A. Les données

Nous avons travaillé sur 2 principaux types de données, des données de transport en commun (TC) ainsi que des données démographiques. Ces jeux de données sont tous constitués d'au moins un élément de type spatial permettant leur visualisation depuis un logiciel SIG.

1. Les données de transport public

Dans le cadre du Bureau National Transport de l'AFNOR ont été mis en place des groupes de travail en vue de normaliser les données relatives au Transport Collectif. Le ministère contribue activement à cette normalisation, car les normes ont pour but de faciliter l'interopérabilité, les échanges et l'utilisation des données dans différentes applications, et donc in fine doivent favoriser le développement des alternatives à la voiture individuelle (le TC, en l'occurrence).
Une information à jour sur l'avancement des travaux concernant ces normes dans le domaine du transport public est disponible à l'adresse suivante : www.normes-donnees-tc.org.
Il existe notamment un profil d'échange XML normalisé pour l'échange de données d'offre de transport public, appelé Neptune, particulièrement utile pour la mise en place de systèmes d'information multimodale fédérant toutes les données de TC sur une région ou un département, et un logiciel libre associé, financé par le ministère, gérant ce profil d'échange : le logiciel Chouette.
Ces derniers mois, la tendance est à une publication sous licence open data de données publiques d'offre de TC. Le format le plus répandu pour cela est d'origine américaine : GTFS, qu'il est possible de convertir au format Neptune, et réciproquement grâce au logiciel Chouette.

Nous avons donc utilisé dans un premier temps ces données open data disponibles sur le web (par exemple pour Rennes, à l'adresse http://data.keolis-rennes.com/) en utilisant l'outil www.chouette.mobi puis des données publiées après le début du stage sur le site www.lepilote.com.

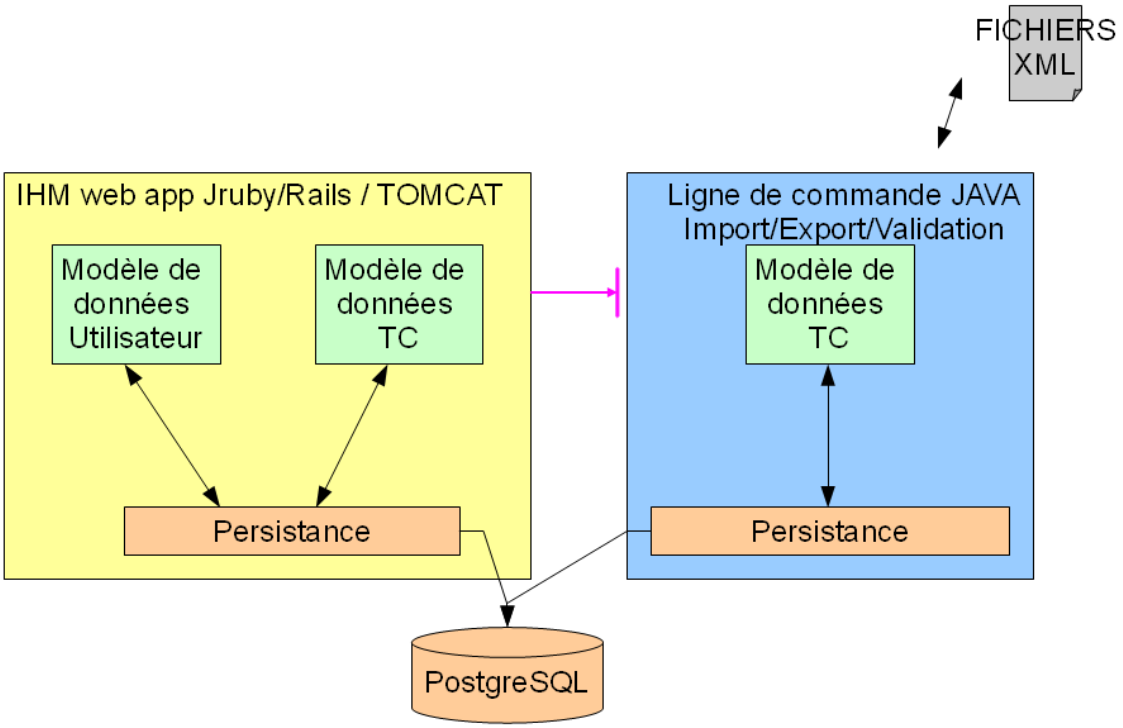
2. Les données démographiques

La seule donnée d'offre de transport en commun ne suffit pas aux besoins des chargés d'études : des données de type démographiques délimités par zone doivent être croisés avec cette offre pour permettre des analyses. Des données de ce type sont disponibles sur le site de l'INSEE, que ce soit par IRIS (découpage en îlots) ou par carroyage (découpage au km et plus récemment 200m), au format Excel dans des tableaux incluant un grand nombre d'attributs variés. Dans notre cas, nous avons utilisé les données de population déjà intégrées et gérées dans une base de données par le pôle Géomatique du CETE.

B. Les logiciels

1. Chouette et ses composants

CHOUETTE est un logiciel libre développé à l'initiative du ministère, dans le but de faciliter l'échange de données d'offre (théorique) de transport collectif, en s'appuyant pour cela sur la norme NFP 99506, dite Neptune, qui spécifie un profil d'échange XML.
Chouette est organisé selon l'architecture suivante :



Les principales fonctions de ce logiciel sont, dans l'ordre :

- valider des données au format XML/Neptune
- visualiser des réseaux TC
- importer / exporter / convertir des jeux de données
- gérer et mettre à jour des données d'offre TC.

Les utilisateurs visés sont les collectivités locales Autorités Organisatrices de Transport (AOT), les exploitants des réseaux TC, et leurs prestataires (bureaux d'étude ou société de services et éditeurs de logiciels), ainsi que les services de l'état, les opérateurs de services d'information, ou les chercheurs...

Le logiciel est utilisable de plusieurs façons :

- en tant qu'application web hébergée gratuitement sur www.chouette-mobi ;
- en tant qu'application web à installer sur votre serveur ;
- en tant que machine virtuelle à installer sur son PC.

Et pour les développeurs :

- en ligne de commande (shell) ;
- en ruby ;
- en tant que librairie java réutilisable.

Pour pouvoir fonctionner, Chouette doit être hébergé sur un serveur : le plus simple est d'utiliser le conteneur Tomcat d'applications java. Les modalités d'installation sont détaillées ci-dessous.

2. QGIS

Le logiciel QGIS est le système d'information géographique (SIG) open-source le plus répandu, il est d'ailleurs largement utilisé au CETE et au ministère.

Un SIG est un système d'information permettant l'interaction avec des données alphanumériques spatialement référencées, autrement dit géo-référencées, ainsi que la production de plans et de cartes. L'usage courant du système d'information géographique est la représentation plus ou moins réaliste de l'environnement spatial en se basant sur diverses formes de données, dont au moins une géométrique.

QGIS fait partie des projets de la Fondation Open Source Geospatial, il gère les formats d'image matricielle (raster) et vectorielles, ainsi que les bases de données. Ce logiciel est plutôt simple d'utilisation même pour une personne sans connaissance particulière des SIG. De plus QGIS permet une visualisation, via différentes couches, des objets PostGIS ; il était clairement le SIG le plus pratique pour notre projet.

3. PostgreSQL et ses composants

PostgreSQL est un système de gestion de base de données (SGBD), c'est le principal SGBD relationnelles open source avec MySQL et SQLite.

Les performances de PostgreSQL permettent de traiter de gros volumes. Une documentation très fournie est disponible en ligne à l'adresse suivante : http://docs.postgresqlfr.org/9.2/ .

L'outil graphique pgAdmin permet une prise en main simplifiée, et grâce à la documentation citée ci-dessus, l'écriture en SQL s'en voit également plus facile à approcher ; cependant, sans connaissance préalable de ce langage, l'écriture de certains scripts peut s'avérer complexe et difficile à débugger.

PostgreSQL dispose d'une extension géographique PostGIS. Cette extension permet de décrire la géométrie des objets gérés dans les tables, de manière à pouvoir les visualiser dans un Système d'Information Géographique (SIG), et à pouvoir faire des requêtes spatiales et topologiques (proximité, inclusion, etc.).

III. MISE EN PLACE DE L'ENVIRONNEMENT DE TRAVAIL

A. Installation de l'environnement de travail

Pour l'installation, nous nous sommes appuyés sur la documentation d'installation de l'application Chouette disponible en ligne sur le site http://chouette.mobi , puis sur le manuel d'installation de postgis.
Il faut commencer par télécharger les composants logiciels qui permettront le bon fonctionnement de l'application, les commandes décrites ci-dessous sont à exécuter dans le terminal sous un système d'exploitation Linux ; dans notre cas nous avons utilisé la version 12.04 LTS de Ubuntu.

Effectuer la commande *apt-get install* suivie des différents packets :

- Plate forme JRE : *openjdk-7-jre X* (X étant la version désirée) ;
- Serveur WEB Tomcat : *tomcat ;*
- Base PostgreSQL : *postgresql* (X.X pour une version précise sinon la dernière version sera installée par défaut) ;
- PostGIS : *postgis ;*
- Proj4 : *proj-bin* et *libproj-dev ;*
- PgAdmin : *pgadmin* ;

Concernant l'installation de QGIS la commande précédente ne fonctionne pas directement, le packet ciblé par défaut étant obsolète. Il faut alors effectuer les commandes suivantes en respectant l'ordre d'exécution :

- *sudo apt-add-repository ppa:ubuntugis/ubuntugis-unstable*
- *sudo apt-get update*
- *sudo apt-get install qgis*

B. Création de la base de données et mise en place de l'application

1. PostgreSQL

Pour commencer, créer un compte applicatif PostgreSQL qui sera le propriétaire de la base via le terminal :

createuser -d -P -U postgres <nom_user> ; ici nom_user = chouette, lui donner pour mot de passe chouette lorsque cela sera demandé et répondre oui à toutes autres questions.

Puis créer une base de données qui lui sera associée :

createdb -E UTF-8 -T template1 -O <nom_user> <nom_base>

2. Chouette

a) Installation

Télécharger les fichiers « chouette2.war » et « command.zip » depuis le site www.chouette.mobi rubrique chouette => version pré-compilées.

Déplacer ensuite l'archive « chouette2.war » dans le dossier */var/lib/tomcatX/webapps,* où X est la version de Tomcat. L'archive va être automatiquement décompressée par le serveur.

Décompresser également la seconde archive dans le dossier */usr/local/opt/chouette-gui-command-V2.*

Continuer par le paramétrage des fichiers en fonction de l'annexe 7.2 du manuel d'installation « Chouette v2.0.2 » disponible de même sur le site www.chouette.mobi. Ces fichiers sont accessibles à partir du dossier */var/lib/tomcatX/webapps/chouette2/* (lors de ce paramétrage, vous devrez créer les dossiers imports/exports/validations aux emplacements spécifiés et veiller à y attribuer les bons droits).

b) Installation (variante)

Pendant le stage, l'application utilisant la version 2.0.2 pré-compilée ayant rencontré des difficultés de fonctionnement, nous avons décidé d'effectuer nous-mêmes la génération de l'application à partir du code source de la version 2.0.3.
Il faut pour cela télécharger plusieurs packets Linux :

- Git : *git* ;
- Plate Forme JDK : *openjdk-7-jdk X ;*
- Maven : *maven ;*
- Jruby : *jruby .*

Puis nous avons suivi le manuel d'installation de l'application, à partir de la rubrique 3.2 qui décrit les différentes manipulations à effectuer pour réussir cette génération. Cela nous a permis de mieux comprendre le fonctionnement interne de l'application, mais également de tester la compréhension de cette partie du manuel et ainsi de soumettre à l'équipe de maintenance du logiciel quelques modifications pour en faciliter l'utilisation.

c) Utilisation

Une fois le paramétrage effectué, relancer le serveur Tomcat pour qu'il prenne en compte les modifications (dans un terminal : */etc/init.d/tomcatX restart* avec X correspondant au numéro de version), puis ouvrir l'application localement via l'url : http://localhost:8080/chouette2, qui correspond à l'IHM de l'application Chouette.

Il faut ensuite créer une nouvelle organisation et un compte de connexion, ce qui nécessite un adresse e-mail (en cas de problème de connexion lié à l'activation du compte, se référer à la partie 4.4.2 du manuel d'installation).

Une fois connecté à l'application Chouette, créer un espace de données, qui sera visible dans la base de données sous forme de schéma. Après la création de cet espace, vous pourrez importer ou exporter des données à votre guise via les onglets correspondants dans l'application.

Ces actions d'import peuvent échouer, si tel est le cas cela vous sera spécifié (actualiser la page et regarder la couleur de la barre de chargement). Ces échecs peuvent être dus à un non respect de la norme Neptune par les fichiers à import, c'est pourquoi il existe également une fonction de validation de fichiers pour pouvoir connaître à l'avance si un fichier est importable ou non dans la base : ce validateur se trouve dans l'onglet « validation » et vous détaillera les erreurs détectées.

3. PostGIS

Ouvrir pgAdmin : vous pourrez ainsi vérifier que le schéma correspondant à l'espace de données créé dans Chouette apparaît bien dans la base , que sa structure correspond au modèle de données métier de Chouette et que les données ont bien été importées si vous avez réalisé l'étape ci-dessus.

Créer, sur l'ensemble de la base, une extension PostGIS de la façon suivante :

- Installer Postgis (apt-get install postgis sous Linux)
- Dans une fenêtre SQL faire : *CREATE EXTENSION postgis;*
- Puis appliquer l'extension sur le/les schéma(s) de son choix : *ALTER EXTENSION postgis SET SCHEMA nom_schema;*

Les fonctions de PostGIS seront alors disponibles dans la base et vous pourrez de ce fait créer des éléments, schémas ou tables, permettant de stocker des données spatiales ainsi que les indicateurs pour plus tard pouvoir faire un lien avec un SIG.

4. QGIS

Pour visualiser les arrêts et tronçons sur un fond de carte, nous avons installé un plugin QGIS appelé « OpenLayers public » (facile à installer depuis le menu d'installation de plugin).
Cette extension permet notamment d'afficher un fond Google Maps (par exemple 'Hybrid' avec photo satellite et voirie).
Cependant, ce fond de plan exige un accès internet et ralentit significativement l'affichage lors du déplacement de la carte par exemple.

Pour appliquer ce fond de carte il vous faut :

- ➔ Cliquer sur l'onglet extension puis sur l'installateur d'extension
- ➔ En rechercher une du nom de OpenLayers public.
- ➔ Une fois l'installation terminée, toujours dans l'onglet extension, cliquer sur le gestionnaire d'extension et activer celle venant d'être installée.

Ensuite, un onglet du nom de l'extension devrait apparaître dans l'onglet d'extension : il vous suffit alors de sélectionner Google Hybrid par exemple.

5. Données démographiques

Les données démographiques sont celles de l'INSEE, qui sont gérées par le pôle Géomatique du CETE Méditerranée qui les met à disposition des autres services. Dans mon cas personnel, j'ai récupéré les données via un *dump* (c.à.d un export des données en ligne de commandes du shell) des tables m'intéressant, celles regroupant les données démographiques des départements d'études, et je les ai rajouté à ma base via la commande *pg_restore* (import en ligne de commandes).

Ces tables sont principalement constituées d'un identifiant unique permettant de les distinguer, d'un attribut geometry permettant la localisation de la zone sous forme de points dans un SIG et d'un attribut numérique décimal correspondant au nombre d'habitants pour l'emplacement (en fait chaque point correspond à un bâtiment et la population en est estimée par interpolation de la population des ilots IRIS).

IV. CONCEPTION DES REQUÊTES

Ce chapitre présente les requêtes que nous avons mises au point : on commence par présenter les indicateurs à calculer, puis le modèle de données sur lequel on travaille, et les requêtes existantes issues du projet Potimart dont nous sommes partis, enfin nous présentons pour chaque indicateur la requête que nous avons mise au point pour le calculer.

Deux scripts SQL ont été créés pour ce projet, ils seront présentés en fin de partie.

A. Les indicateurs à calculer

Le Certu a proposé un cahier des charges initial des indicateurs à calculer au groupe de travail constitué pour ce projet de BD TC. Les participants au groupe (Certu, Service d'observation et étude statistiques transport, CETEs - dont le service géomatique AGIL et la mission MIM du CETE Med) ont fait évoluer la définition des indicateurs. Les indicateurs envisagés seraient calculés progressivement en fonction de la disponibilité des données :

1.Données arrêts XY uniquement :

➔ calcul pour chaque arrêt :
- de la population résidant à moins de 300m ;
- des emplois à moins de 300m (les données d'emploi étant plus complexes à obtenir, cet indicateur sera calculé dans un 2ème temps).

2. Données arrêts XY avec lignes :

➔ calcul pour chaque ligne :
- de la population résidant à moins de 300m d'un arrêt de la ligne;
- des emplois à moins de 300m d'un arrêt de la ligne ;
- de la sinuosité de la ligne (les tracés de ligne n'étant en pratique pas disponibles, cet indicateur sera calculé dans un 2ème temps).

➔ calcul pour chaque arrêt :
- de la population accessible en trace directe depuis cet arrêt (on aurait la même valeur pour tous les arrêts d'une ligne, sauf les arrêts desservis par plusieurs lignes, où on sommerait) ;
- idem pour les emplois.

➔ calcul pour une zone (un îlot?) :
- du nombre de lignes situées à moins de 300m de l'îlot.

3. données arrêts + lignes + horaires :

➔ calcul pour chaque arrêt :
- de la fréquence des passages à l'arrêt (en distinguant éventuellement "moyenne journée" / "heure de pointe" / "heure creuse") ;
- de l'amplitude (1er horaire; dernier horaire; durée entre les 2) ;
- emplois et population desservis en x minutes (complexe à calculer, sera traité dans un 2ème temps) ;
- temps d'accès TC jusqu'à un point central de la ville (ex : la gare principale; l'hôtel de ville; ...) (complexe à définir, sera donc calculé dans un 2ème temps).

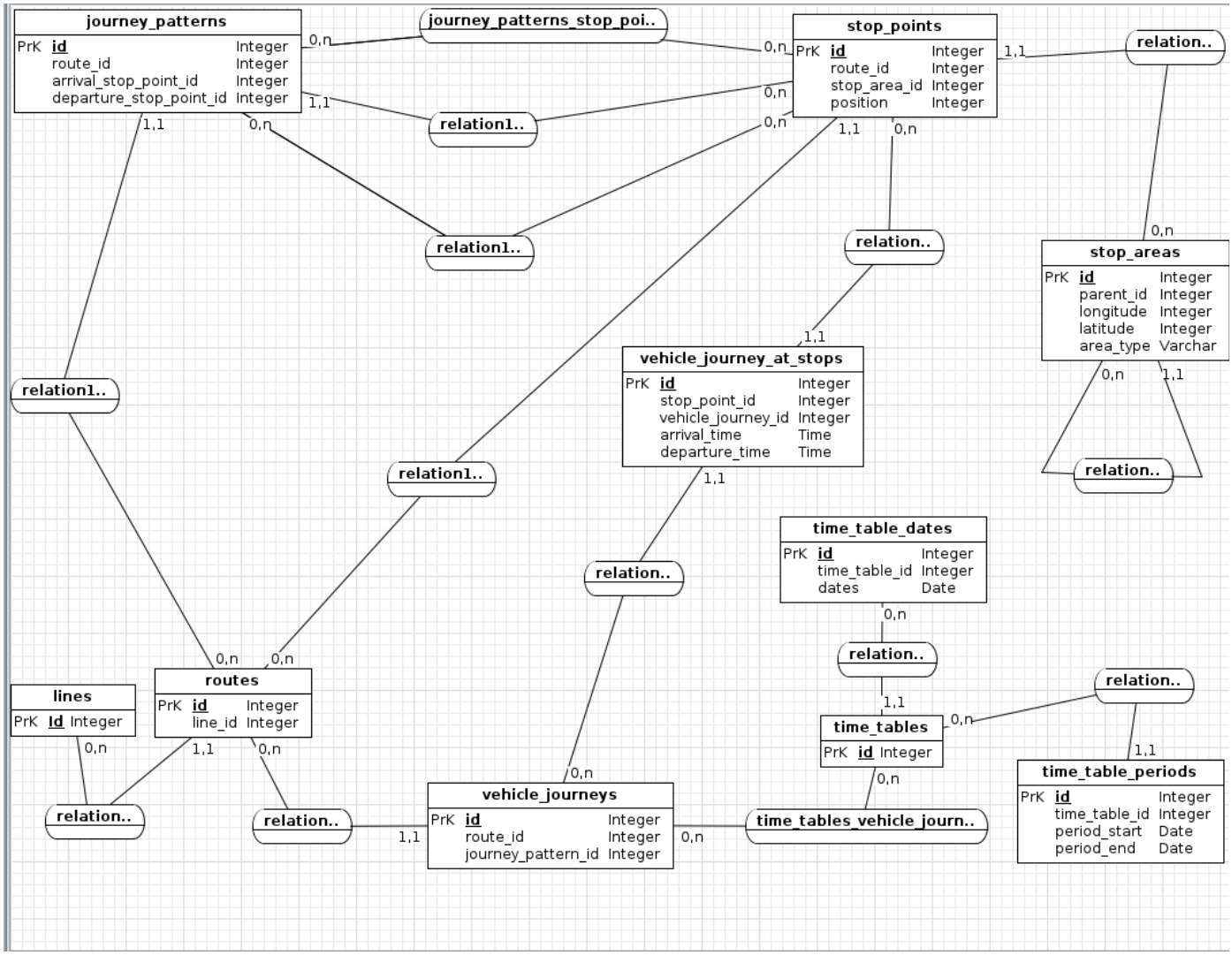
➔ calcul pour une zone (* à préciser dans un 2ème temps) :
- nombre d'arrêts accessibles en moins de x minutes d'un point central de la ville (ex : la gare principale; l'hôtel de ville; …) ;
- cartes isochrones.

Pour « peupler » cette Base de Données, plusieurs sources de données sont envisageables :
- les arrêts et lignes d'Open Street Map ;
- les données open data publiées par certaines collectivités Autorités Organisatrices de transport (Rennes, Bordeaux, Nantes, Bouches-du-Rhône...) ;
- la BD Nord-Pas-de-Calais constituée par la DREAL Nord-Pas-de-Calais et gérée par le CETE Nord-Picardie ;
- suite à la sollicitation officielle faite par le ministère auprès des Collectivités Locales, dans le cadre de ce projet, les données que fourniraient les Régions et Départements qui ont mis en place des Systèmes d'Information Multimodales sur leur territoire fédérant toutes les données TC, mais n'ayant pas forcément publié ces données en open data.

En pratique pour le stage, le travail portera d'abord sur les données open data, avec des jeux de données incluant les horaires (et pas seulement des couches SIG d'arrêts ou de lignes).

B. Présentation de la base et du modèle métier TC

Le Modèle Conceptuel de Données de la figure ci-après donne une vision d'ensemble des données TC ; il contient uniquement les tables (et leurs attributs) d'un schéma de données Chouette nous ayant été utiles durant le stage:



Un n°ligne correspond à l'attribut *lines.id* ou à un des attributs en clé étrangère de la base pointant vers cet attribut.

Un n°arrêt correspond à l'attribut *stop_areas.id* ou à un des attributs en clé étrangère de la base pointant vers cet attribut.
Un n°période correspond à l'attribut *time_table_period.id* ou à un des attributs en clé étrangère de la base pointant vers cet attribut.
Un n°calendrier correspond à l'attribut *time_tables.id* ou à un des attributs en clé étrangère de la base pointant vers cet attribut.
La table vehicle_journey_at_stops définit les modalités des passages de TC aux arrêts.

Les principales notions du modèle de données TC de Chouette (qui sont échangées dans le profil Neptune) sont les suivantes :
- réseau (network)
- transporteurs (company)
- groupe de lignes (group of lines)
- ligne (line)
- séquence d'arrêt (route)
- mission (journey_patterns)
- course / horaires (journey)
- arrêts (stops)
- correspondances (connection_link)
- accès (access)
- calendrier d'application (time_table)

Le Modèle Conceptuel de Données (MCD) ci-dessus montre comment s'articulent ces objets :
- une ligne est composée de séquences d'arrêts (typiquement l'aller et le retour dans le cas le plus simple) ; une séquence d'arrêts comprend plusieurs missions (dans le cas le plus simple, une seule : la mission passe par tous les arrêts, mais il peut y avoir des missions qui sautent certains arrêts, pour distinguer des services « express », « direct » ou « semi-direct »). Les courses qui sont rattachées aux missions d'une ligne définissent les horaires de passage aux arrêts (vehicle_journey_at_stops) et sont valables pour des calendriers qui définissent les jours où s'appliquent les horaires ; ces calendriers contiennent une liste de dates et de périodes (entre deux dates), croisés avec les jours de la semaine où les appliquer.

La notion d'arrêt elle-même n'est pas si simple. Les horaires de passage s'appliquent à des arrêts « topologiques », correspondant à un numéro dans une séquence d'arrêts, pour une ligne donnée ; ces arrêts sont décrits dans la table StopPoints. Les arrêts sur le terrain, ou arrêts « physiques » correspondant à des points d'embarquement pour le bus ou le car, ou des quais pour le métro ou le train, sont stockées dans une autre table, StopAreas. Il y autant de StopPoints que de lignes qui passent par un même arrêt physique StopArea. Cette table StopAreas contient également des arrêts dits « commerciaux », qui sont le regroupement d'arrêts physiques (typiquement 2 arrêts qui se font face dans la même rue, pour l'aller et le retour) ; les arrêts commerciaux peuvent également être regroupés en « pôles » (stop place).

C. travail initial : projet Potimart

Potimart est un projet de R&D visant à développer des solutions open source pour le traitement géomatique de données transport (http://www.mobigis.fr/work-view/potimart/).

La 1ère phase en 2007-2008 a permis de produire des requêtes SQL de calcul d'indicateurs relatifs d'une part aux réseaux routiers (calculs d'impasses et de mailles), d'autre part aux réseaux de TC : nombre de courses passant à un arrêt dans une journée, vitesse moyenne de tronçons de ligne (disponibles sur https://github.com/Potimart/Potimart).

Ces requêtes ont été développées avec la version 1 du logiciel Chouette et une version antérieure de PostGIS. Notre travail a donc d'abord consisté à adapter ces requêtes aux nouvelles versions de Chouette et PostGIS.
Références : http://www.cete-mediterranee.fr/tt13/www/article.php3?id_article=118 et http://www.cete-mediterranee.fr/tt13/www/article.php3?id_article=192

D. Indicateurs des points d'arrêt

Les indicateurs concernant les points d'arrêt seront calculés d'abord pour les arrêts physiques ; ils pourront dans un 2ème temps être calculés pour les arrêts commerciaux (ce qui peut s'avérer plus pratique si on veut visualiser la couche de points sur une carte dans un SIG), voir pour les pôles d'échange, à un niveau plus agrégé.

Ils seront stockés dans la table stop_area_geos contenant en plus des indicateurs ci-après :

- Le numéro d'arrêt
- Le nom d'arrêt
- L'objet géométry de type point
- Le type de l'arrêt

1. Amplitude des horaires

a) Description

L'amplitude horaire à un arrêt est la différence de temps, soit une durée, entre l'heure de passage du premier TC et l'heure de passage du dernier TC, pour un jour donné. L'amplitude est donc calculée sur la base de toutes les lignes qui passent par l'arrêt. Nous chercherons à déterminer l'amplitude pour chacun des arrêts en fonction de la ligne et du calendrier d'application, puis pour un jour donné.

L'amplitude horaire des passages de TC vaut donc :

heure du dernier passage - heure du premier passage.

b) Méthode de calcul

Pour débuter, nous avons commencé par écrire une requête SQL permettant de faire la sélection de chaque triplet d'attributs n°ligne / n°arrêt / n°calendrier présent dans la base. La requête calcule l'amplitude horaire pour chaque arrêt de chaque ligne passant par l'arrêt pour chaque calendrier, ainsi que les horaires du premier et du dernier passage.

Cette requête utilise les tables suivantes :

vehicle_journey_at_stops ; vehicle_journeys ; stop_points ; time_tables ; time_tables_vehicle_journeys ; routes

Le résultat de cette requête est stocké dans la table : *indicateurs.stop_area_line_amplitude.*

Dans un deuxième temps, l'objectif est de compléter la colonne Amplitude de la table *indicateurs.stop_area_geos* qui contient la couche de points d'arrêt. Pour cela, nous avons créé une fonction de profil :

select AreaAmplitudeByDate(id_area , date, nom_schéma) ;

Cette fonction calcule l'amplitude d'un arrêt, dont l'identifiant est passé un premier paramètre, toutes lignes confondues, en fonction d'une date (format 'YYYY-MM-DD'), passée en second paramètre. Comme on veut pouvoir produire cet indicateur pour plusieurs jeux de données, un troisième paramètre fournit le nom du schéma de BD sur lequel les données sont stockées, par exemple dans notre cas le schéma *star*.

Cette fonction retourne l'amplitude horaire au format *integer* représentant le nombre de minutes.

Les tables utilisées pour cette fonction sont les suivantes :

vehicle_journeys, vehicle_journey_at_stops, stop_points, time_tables, time_table_periods, time_tables_vehicle_journeys.

Une fonction similaire a également été écrite pour les arrêts commerciaux :

AreaComAmplitudeByDate(bigint, date,text) ;

2. Nombre de passages un jour donné

a) Description

Nous cherchons ici à calculer combien de passages de TC à chaque arrêt sont effectués pour une journée donnée. Nous calculerons comme ci-dessus plusieurs nombres de passages pour chacun des arrêts, en fonction en fonction de la ligne et du calendrier d'application, puis pour un jour donné.

b) Méthode de calcul

Pour effectuer ce calcul, nous avons écrit une requête permettant la sélection de chaque triplet (ligne, arrêt, calendrier) présent dans la base et effectuant le calcul du nombre de passages pour ce même triplet.

Des indicateurs plus fins sont également calculés, comme souhaité dans le cahier des charges du Certu : pour distinguer les différences d'affluence dans une journée, nous effectuons aussi des calculs pour un découpage en tranches horaires (heures creuses 20h-7h30, heures de pointe 7h30-9h30 & 16h30-20h, moyenne journée 9h30-16h30).

Chaque triplet doit avoir un nombre de passages pour chaque type d'horaire. De ce fait, il faut effectuer des jointures externes entre la relation stop_areas décrivant les arrêts et la relation VehicleJourneyAtStop décrivant les horaires pour ne pas omettre un arrêt non desservi pendant un type d'horaire. Il faut utiliser, dans la requête, une relation par type d'horaire souhaité, cependant n'ayant personnellement pas réussi à mettre en place plusieurs jointures externes sur une même relation (ici celle des arrêts), nous avons écrit autant de requêtes que de types de tranche horaire souhaités (ici trois).

Ces requêtes utilisent toutes les mêmes tables qui sont :

routes, vehicle_journeys , vehicle_journey_at_stops, stop_points, time_tables , time_tables_vehicle_journeys

Pour effectuer la jointure externe la table *vehicle_journey_at_stops* est utilisée deux fois. Le résultat de ces requêtes est stocké sur la table *indicateurs.stop_area_line_nbpas.*

Pour les calculs des indicateurs par tranche horaire, nous avons créé une table (*indicateurs.stop_area_line_nbpas*) donnant le détail des résultats pour chaque triplet n°ligne / n°arrêt/ n°calendrier du schéma de données traité. En revanche, cette table n'est pas visible depuis un SIG comme l'est la table *stop_area_geos*. Il serait bien sûr possible d'ajouter des colonnes à cette dernière table pour les indicateurs de nb de passage par tranche horaire.

Le profil des fonctions de calcul des nombres de passage aux arrêts est le suivant (elles prennent toutes en paramètres un n°arrêt ainsi qu'une date) :

select AreaNbPasByDate(id_area , date, nom_schéma) ;

AreaNbPasMJByDate : pour le nombre en « moyenne journée »,

AreaNbPasHCByDate : pour le nombre en « heure creuse »,

AreaNbPasHPByDate : pour le nombre en « heure de pointe ».

La première fonction calcule et renvoie (au format *integer*) le nombre de passages à un arrêt pour une date, toutes tranches horaires confondues ; il faut également lui passer le nom du schéma source en paramètre au format texte, par exemple pour notre projet *star*.

Cette fonction reprend le script SQL écrit dans POTIMART et est réadaptée au nouveau schéma de la base Chouette V2. Les autres fonctions ont exactement le même fonctionnement, mis à part que chacune compte uniquement le nombre de passages sur une tranche horaire particulière au lieu de toute la journée.

Ces fonctions utilisent toutes les même tables :

vehicle_journeys, vehicle_journey_at_stops, stop_points, time_tables_vehicle_journeys, routes, time_table_periods, time_tables

Comme pour les requêtes précédentes,t la table *vehicle_journey_at_stops* est utilisée deux fois dans le cadre des jointures externes.

De même, des fonctions ont été développées pour les arrêts commerciaux :

AreaComNbPasByDate(bigint, date, text) ;

AreaComNbPasHCByDate(bigint, date, text) ;

AreaComNbPasHPByDate(bigint, date, text) ;

AreaComNbPasMJByDate(bigint, date, text) ;

3. Nombre d'habitants à proximité

a) Description

Nous cherchons ici à calculer combien de personnes vivent à proximité de chaque arrêt du réseau. La distance seuil pour savoir si un arrêt est proche d'une habitation a été fixée à 300 mètres ; elle pourrait bien sûr être paramétrable.

b) Méthode de calcul

Le script doit produire une table pouvant contenir un n° d'arrêt, un objet geometry ainsi qu'une valeur représentant le nombre d'habitants : *indicateurs.stop_area_hab* .

Pour compter le nombre d'habitants autour d'un point d'arrêt, il suffit de récupérer les habitations à moins de 300 mètres de ce point, et de sommer la population (estimée) affectée à chaque habitation. Dans notre base, chaque habitation est modélisée comme un point, et l'opération spatiale à effectuer est donc une intersection entre les points d'habitations de la table issue du recensement Insee et le disque de 300 m de rayon autour du point d'arrêt.

Pour cela, il faut disposer d'une table contenant les arrêts avec la zone de 300 mètres autour de chaque arrêt (dans un objet geometry de type polygone) :

indicateurs.stop_area_hab

Cette table est issue de la table stop_area_geos utilisée dans le projet POTIMART. Comme les géométries des habitations dans la table de population ne sont pas dans le même système de référence géographique que les points d'arrêt (qui sont en coordonnées GPS : WGS84), il faut effectuer une transformation via la fonction postgis *st_transform* avant de pouvoir effectuer la requête spatiale d'intersection. Ensuite, il faut créer le disque de 300 mètres autour de l'arrêt : pour cela on utilise la fonction postgis *st_buffer*. Nous insérons ensuite le résultat de cette requête dans la table *indicateurs.stop_area_geos*.

Il faut ensuite une seconde requête pour calculer, pour chaque arrêt, l'attribut correspondant au nombre d'habitants en utilisant la fonction *st_intersects.* Cette fonction vérifie si des attributs de type géométry se croisent (ici les points représentant les habitations dans la table de population, et les disques autour d'un arrêt). Pour l'ensemble des résultats positifs, nous faisons le calcul de la somme du nombre d'habitants, via la fonction *sum* puis nous mettons à jour la table de stockage de ces éléments avec les résultats de la requête.

Cette requête utilise les tables :

indicateurs.stop_area_hab et *zz_pop.ff_d35*

Nous avons dû séparer les deux requêtes (calcul du polygone autour de chaque point d'arrêt, et calcul du nombre d'habitants). En effet l'imbrication de ces deux requêtes en une seule avait pour conséquence de fortement ralentir la durée d'exécution.

E. Indicateurs des tronçons de lignes

Un tronçon de ligne correspond au chemin entre deux arrêts successifs d'une même ligne, ce tronçon ne prendra pas en compte la sinuosité de la route mais sera le chemin à vol d'oiseau entre les deux arrêts.

Ces tronçons seront représentés dans la table *service_links* contenant en plus des indicateurs exposé ci-après :

- Un identifiant
- les numéros d'arrêts d'arrivés et de départs
- les numéros de ligne, de route, et de mission
- Un objet géométry de type (poly-)ligne

Ce qui identifie un tronçon est le triplet (arrêt départ, arrêt arrivée, itinéraire) : deux tronçons peuvent donc être superposés : typiquement cas de deux lignes qui passent par les 2 mêmes arrêts successifs.

1. Vitesse moyenne

a) Description

La vitesse moyenne sur un tronçon de ligne est le rapport entre durée moyenne en empruntant la ligne de TC et distance à vol d'oiseau pour deux arrêts consécutifs d'une ligne, quelle que soit la date (on ne tient pas compte des calendriers, pour le calcul ; il serait possible de le faire).

La vitesse sera donc inférieure à celle qu'on obtiendrait si on disposait du tracé des lignes, forcément plus long qu'une ligne brisée. Néanmoins c'est bien le temps de parcours entre 2 points qui est pertinent pour décrire le service, donc le calcul sur la base d'une distance à vol d'oiseau est assez pertinent ; il n'est pas évident qu'un calcul plus réaliste sur la base du tracé effectif des lignes apporterait quelque chose.

b) Méthode de calcul

Nous avons donc commencé par produire la table des tronçons de ligne, d'où l'écriture d'une requête SQL produisant tous les couples d'arrêts successifs d'une séquence d'arrêts appartenant à une même mission.

En effet, rappelons qu'une ligne comprend plusieurs séquences d'arrêts (routes en anglais), typiquement deux (aller et retour), mais que les horaires sont définis au niveau des missions et pas des séquences d'arrêt, car dans le cas général il peut exister des courses qui ne s'arrêtent pas à tous les arrêts, donc chaque séquence d'arrêt peut comprendre plusieurs missions (journey pattern en anglais), et donc les calculs de vitesse moyenne, qui se font à partir des horaires, impliquent de définir les tronçons à partir des missions. Nous ajoutons également deux attributs pour chaque tronçon : sa distance et la durée moyenne nécessaire pour parcourir le tronçon.

Cette requête utilise les tables suivantes :

indicateurs.service_links, stop_points, vehicle_journeys, vehicle_journey_at_stops, indicateurs.stop_area_geos, stop_areas

Pour calculer la distance, nous utilisons la fonction de PostGIS *st_distance* qui retourne l'écart à vol d'oiseau entre deux objets geometry. Pour que la distance soit retournée en mètres, il faut faire les calculs dans un référentiel spatial adéquat (Lambert 93 par exemple, en tout cas pas WGS84 qui retournerait un résultat en degrés) : pour cela, nous utilisons la fonction *st_transform*.

Pour calculer la durée des trajets, nous avons développé une fonction de profil :

select DiffTime(time without time zone, time without time zone) ;

Cette fonction calcule l'écart entre deux horaires successifs et le renvoie au format *time without time zone.* Cette fonction gère les divers cas problématiques, notamment les trajets chevauchant deux journées ou de durée nulle. Le premier paramètre est l'horaire de départ, le deuxième est l'horaire d'arrivée à l'extrémité du tronçon.

Il suffit ensuite d'appeler la fonction dans la requête en lui passant les horaires de départ ainsi que les horaires d'arrivée et de faire la moyenne de celles-ci pour chaque couple de *stop_area*.

Reste à insérer le résultat dans la table *indicateurs.stop_area_distance,* contenant les couples d'arrêt ainsi que la distance et la durée les séparant. La table peut être créée dynamiquement, c'est à dire directement avec les résultats de la requête, pour être certain de sa structure, mais dans ce cas il ne faut pas oublier de rajouter les contraintes par la suite.

Nous avons ensuite créé une fonction de profil :

select avgSpeed(double precision, interval) ;

Elle prend en paramètres une distance et une durée, elle convertit les attributs passés en paramètres afin qu'ils soient de types convenables pour un calcul de vitesse. Puis ce calcul est effectué et la fonction renvoie donc une vitesse moyenne de type *double precision.*

Cette vitesse est ensuite stockée dans une table contenant la géométrie du tronçon (un segment, donc) afin de pouvoir la faire apparaître dans un SIG. Cette table est adaptée de la table *service_links* du projet POTIMART que nous avons remplie à l'aide de la fonction *insert_links* du même projet, que nous avons cependant adaptée avec le profil suivant :

select insert_links(journey_patterns_id, schéma_cible, schéma_source) ;

La fonction prend un n° de mission en paramètre et nous avons ajouté comme paramètre le nom du schéma cible où nous voulons insérer les données dans la table *service_links*, et également le nom du schéma source où sont stockées les données initiales ; ces deux noms de schémas sont au format texte.

Cette fonction utilise les tables :

vehicle_journeys, vehicle_journey_at_stops, stop_points, stop_areas , routes, lines

et écrit sur la table *indicateurs.service_links.*

2. Nombre d'habitants moyen par arrêt pour une ligne

a) Description

Le nombre d'habitants moyen est un indicateur relatif à une ligne de TC (et donc applicable aussi aux tronçons de ligne) : c'est la somme du nombre d'habitants vivant à proximité d'un des arrêts divisé par le nombre d'arrêts de la ligne.

En toute rigueur, cette méthode de calcul est erronée si des arrêts sont à moins de 600 mètres les uns des autres, car elle fait alors des doubles comptes dans les zones d'intersection entre les disques à moins de 300 m de 2 arrêts consécutifs. Nous n'avons toutefois pas eu le temps de compléter l'implémentation pour corriger ce défaut.

b) Méthode de calcul

Nous créons une fonction de profil :

select avgNbHabByLine(id_line, schéma_cible1, schéma_cible2) ;

qui prend un numéro de ligne en paramètre, le nom du schéma stockant les indicateurs puis le nom de celui stockant les données TC, les deux au format texte. Cette fonction calcule la valeur moyenne du nombre d'habitants de la ligne, en sélectionnant dans la table stockant le nombre d'habitants à proximité de chaque arrêt. La fonction effectue ce calcul en sélectionnant les arrêts appartenant à la ligne et en utilisant la fonction *avg* sur l'attribut stockant le nombre de personnes vivant à proximité. Le type du résultat de la fonction est un *integer*.

La fonction requiert les tables suivantes :

indicateurs.stop_area_hab, stop_points, routes

Une fonction similaire pour les arrêts commerciaux a été développée :

select avgNbHabByLineCom(id_line, schéma_cible1, schéma_cible2) ;

La seule différence notable est que cette fonction utilise également la table *stop_areas* pour cibler les arrêts commerciaux.

F. Test avec les données open data de Rennes

Des indexes relationnels et des indexes spatiaux ont été générés sur la plupart des tables utilisées, ils sont créés directement dans le script SQL, lors des déclarations des tables.

Le nom du schéma correspondant à l'espace de données créé via l'application Chouette sera ici star puisque nous avons travaillé avec les données open data du réseau de TC de Rennes, qui s'appelle STAR.

Nous avons pu tester les différentes requêtes décrites ci-dessus avec les données open data de Rennes. Nous avons constaté que ces données avaient, pour chacune des manipulations, un résultat apparemment correct et exploitable. En effet, aucune erreur provenant des données n'a été détectée et nous avons même pu affiner nos requêtes et les tester pour obtenir les résultats les plus justes possibles. Nous avons également pu croiser sans erreur ces données avec les données de population démographiques.

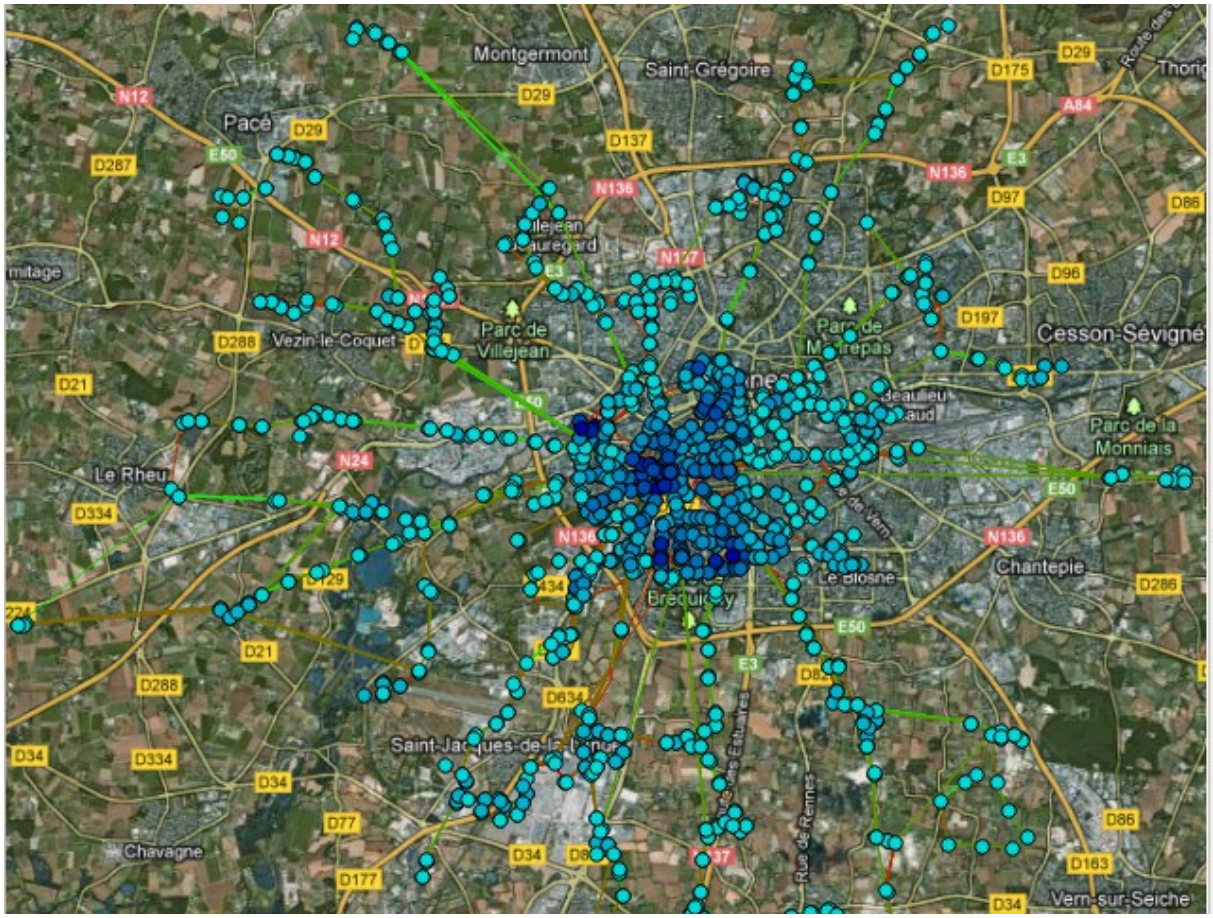
Pour faire apparaître des données stockées dans la base via le logiciel QGIS, il faut :

- ➔ Ouvrir QGIS.
- ➔ Ajouter une nouvelle couche PostGIS (onglet couche => Ajouter une nouvelle couche PostGIS), vous devrez pour cet ajout choisir une connexion, à cette étape aucune n'est préalablement créée.
- ➔ Cliquer sur « Nouveau » puis rentrez les paramètres relatifs à votre base, confirmer.

Dans « Connecter » devrait apparaître la table stop_area_geos de la base (si vous l'avez créé comme dans le script POTIMART proposé précédemment) et/ou toutes les tables contenant un attribut spatial.

Il est bien sûr plus simple d'ouvrir directement les fichiers Shapefile que nous générons pour chaque jeu de données avec QGIS ou n'importe quel autre SIG, le but étant que l'utilisateur final (chargé d'études transport) n'aie pas accès à la Base PostgreSQL.

Voici un exemple des résultats obtenus avec le logiciel QGIS : le code couleur des arrêts correspond au nombre d'habitants à proximité, à la vitesse moyenne des tronçons de ligne:



Les données de Rennes, de très bonne qualité, nous ont permis de mettre au point les scripts.

Nous avons ensuite à la fin du stage exécuté ces scripts sur les données open data disponibles et importables dans CHOUETTE.

G. Réutilisation des résultats

Après l'acquisition d'un nouveau jeu de données NEPTUNE ou GTFS, importer ces données via l'application Chouette pour pouvoir calculer les différents indicateurs.

L'ensemble des requêtes sont regroupées dans deux scripts SQL : *calcul_indicateurs.sql* et *calcul_indicateurs_com.sql* .

Le script *calcul_indicateurs.sql* crée un nouveau schéma (de nom paramétrable) contenant l'ensemble des tables stockant les indicateurs

- *stop_area_distances* contenant chaque couple d'arrêts de départ et d'arrivé des tronçons de ligne ainsi que la distance et la durée les séparant.
- *stop_area_hab* contenant chaque arrêt ainsi que la zone acceptable d'habitation aux alentours et le nombre d'habitant y résidant
- *stop_area_line_nbpas* contenant chaque triplet n°arret / n°ligne / n° calendrier ainsi que le nombre de passage respectif pour chaque type d'horaire.
- *stop_area_line_amplitude* contenant chaque triplet décrit ci-dessus et l'amplitude associée ainsi que l'heure de premier et l'heure du dernier passage.

ainsi que les 2 tables contenant les indicateurs et faisant la liaison avec les couches SIG et les remplit avec les valeurs des indicateurs adéquates pour les arrêts physiques :

- *stop_area_geos* pour les arrêts physiques
- *service_links* pour les tronçons de lignes

Il crée bien sûr aussi les fonctions de calcul et d'insertion de ces indicateurs dans les tables.

Le script *calcul_indicateurs_com.sql* effectue les mêmes actions que le script précédent mais pour les arrêts commerciaux uniquement, y compris les tronçons de ligne ne concernent que ce type d'arrêt.

Ces deux scripts doivent être adaptés à chaque jeu de données (qui s'applique à chaque fois sur un schéma différent en BD). Nous avons donc mis au point un script shell *indicateurs.sh* permettant le paramétrage de ces scripts SQL, leur exécution ainsi que la création des fichiers SIG au format Shapefile associé aux indicateurs venant d'être créés, ils le seront dans le répertoire courant.

Ce script *indicateurs.sh* doit lui-même bien sûr être paramétré avec les bons noms correspondant au jeu de données à traiter (nom du schéma de données, nom du schéma, etc. voir les commentaires de ce script). Pour exécuter ce script, il suffit de taper dans un terminal Linux :

./indicateurs.sh

Pour pouvoir disposer en permanence des méta-données relatives à ces schémas, il faut paramétrer la table *info_schema* en suivant les commentaires y étant associés. Cette table doit être paramétrée manuellement avant chaque lancement de script, elle se trouve en début des scripts SQL.

V. CONCLUSIONS ET SUITES À DONNER

A. En résumé

Dans l'ensemble, le travail réalisé a permis de créer des scripts permettant de calculer de nouveaux indicateurs relatifs aux réseaux de TC. Ces indicateurs ont pu être visualisés via un logiciel SIG, et nous avons ainsi pu découvrir des erreurs sur différents jeux de données et transmettre aux personnes concernés les informations sur ces erreurs afin de corriger les données.

Nous avons également pu améliorer l'utilisation de l'application Chouette, puisque ayant nous-mêmes travaillé avec cette application et ayant dû l'installer, nous avons remarqué des ambiguïtés, ce qui a permis de compléter le manuel d'installation de l'application afin qu'il soit plus clair.

Ce stage m'a également permis de découvrir l'univers professionnel et diverses notions qui seront par la suite un atout pour moi telles que les SIG et les données spatiales ainsi que le SGBD PostgreSQL et ses composants en incluant également certaines notions SQL.

B. Perspectives

Pour la suite du projet, notre prototype sera transféré au service Géomatique AGIL du CETE Méditerranée, qui reprendra la base de données et fera évoluer le cas échéant les scripts (en fonction d'éventuelles corrections ou du besoin de calculer d'autres indicateurs).

En mode production, le CETE gérerait donc une BD Postgres et importerait au fil de l'eau des données à mesure de leur disponibilité puis générerait les fichiers SIG correspondants et les mettra à disposition sur un site intranet (réservé aux agents du ministère, notamment pour les données open data), et les enverrait également pour information et retour éventuel au producteur des données. La BD d'indicateurs TC serait ainsi complétée progressivement, sur un mode itératif (les scripts de calcul d'indicateurs évolueront certainement).

Au schéma des indicateurs associés à chaque jeu de données devra être ajouté une table contenant tous les élements décrivant comment les données ont été produites :

- source de données (texte libre indiquant l'url du site open data ou les références du fournisseur des données, la date de récupération des données)
- la date d'exécution de la requête et les paramètres de cette requête (notamment le jour utilisé pour calculer le nombre de passages aux arrêts)

Cette table permettra de générer la fiche de méta-données associées à chaque lot de fichiers SIG. Les géomaticiens du service AGIL incluent systématiquement une telle fiche à tout jeu de données qu'ils produisent (en bons professionnels).

A terme, si cette base de données nationale des indicateurs de TC est mise en place par le ministère, il faudra mettre en conformité les noms des tables et des attributs avec le modèle de données des lieux d'arrêt en cours d'élaboration par l'Agence Française de l'Information Multimodale et de la Billettique. Cela ne concernera que les points d'arrêt (pas les tronçons) et devrait être très simple : les arrêts physiques sont appelés Zones d'Embarquement et le seul attribut avec lequel faire le lien serait l'identifiant national.

Dans la mesure où l'application Chouette permet de traiter de manière uniforme des données open data publiées au format GTFS et des données au profil NEPTUNE, nous préconisons de travailler autant que possible avec ces données complètes d'offre TC, qui incluent les horaires et permettent ainsi de calculer des indicateurs assez riches.

Néanmoins, lorsque les données mises à disposition se limiteront aux informations concernant les arrêts et les lignes, il sera possible de calculer uniquement les indicateurs de population desservie. Néanmoins, les données TC publiées au format SIG en open data sont pour l'instant sous des formes très diverses, ce qui implique de les analyser au cas par cas, pour voir comment les transformer afin de les traiter ensuite par une seule requête, ou si il est nécessaire d'adapter une requête spécifique pour les données. Cette hétérogénéité des données SIG augmente significativement le temps de traitement et donc son coût. Une alternative serait d'utiliser les données disponibles sur OpenStreetMap : elles sont certes incomplètes et non officielles, mais couvrent la France entière dans un format assez homogène et permettraient de produire rapidement à peu de frais les indicateurs souhaités.

Bien sûr, d'autres indicateurs en lien avec des données sur les emplois, le tourisme ou encore la qualité de service (retards...) et la fréquentation des usagers pourrait être développés par la suite, il faudra cependant disposer de données proprement formatées et ne comportant aucune erreur, c'est pourquoi là encore l'utilisation de l'application Chouette, qui permet de valider les données, est préconisée.

VI.ANNEXE : RÉFÉRENCES

Les résultats de cette étude sont disponibles sur le site web: http://www.cete-mediterranee.fr/tt13/www/

Contexte :
www.predim.org
www.cete-mediterranee.fr/tt13
http://www.normes-donnees-tc.org/
http://www.chouette.mobi/spip.php?rubrique61
http://www.chouette.mobi/documents/Donnees-Neptune.pdf
site Rennes open data
site Lepilote

outils :
www.chouette.mobi
postgis.org
posgresql.org
qgis.org
projet potimart
http://www.cete-mediterranee.fr/tt13/www/article.php3?id_article=296&var_recherche=s%E9mantique
et www.potimart.org
http://www.cete-mediterranee.fr/tt13/www/recherche.php3?recherche=potimart
github.com/potimart

Bibliographie:
http://docs.postgresqlfr.org/9.2/
http://postgis.net/stuff/postgis-2.0.pdf
http://ubuntu-fr.org/