

Importation, analyse et visualisation de données d’enquête dans un modèle multidimensionnel de trajectoires de vie

Plan de test

Etudiants :  
BENKAOUR Salwa

FLEURY Pierre

JORDAN Célia

Enseignant :

PELLIER Damien

Référent :

GENSEL Jérôme

Table des matières

[Plan de test 1](#_Toc106282284)

[Introduction 4](#_Toc106282285)

[Objectifs et méthodes 4](#_Toc106282286)

[Documents de référence 4](#_Toc106282287)

[Guide de lecture 4](#_Toc106282288)

[Maitrise d’œuvre 4](#_Toc106282289)

[Maitrise d’ouvrage 4](#_Toc106282290)

[Concepts de base 4](#_Toc106282291)

[Test fonctionnels 4](#_Toc106282292)

[Pour chaque scenario 4](#_Toc106282293)

[Identification 5](#_Toc106282294)

[Description 5](#_Toc106282295)

[Contraintes 5](#_Toc106282296)

[Dépendances 5](#_Toc106282297)

[Procédure de test 5](#_Toc106282298)

[Test d’intégration 5](#_Toc106282299)

[Pour chaque test d’intégration 5](#_Toc106282300)

[Identification 5](#_Toc106282301)

[Description 5](#_Toc106282302)

[Contraintes 5](#_Toc106282303)

[Dépendances 5](#_Toc106282304)

[Procédure de test 5](#_Toc106282305)

[Test unitaires 5](#_Toc106282306)

[Pour chaque test unitaire 5](#_Toc106282307)

[Identification 5](#_Toc106282308)

[Description 6](#_Toc106282309)

[Contraintes 6](#_Toc106282310)

[Dépendances 6](#_Toc106282311)

[Procédure de test 6](#_Toc106282312)

[Vérification de la documentation 6](#_Toc106282313)

[Annexes 6](#_Toc106282314)

[Glossaire 6](#_Toc106282315)

[Références 6](#_Toc106282316)

[Index 6](#_Toc106282317)

## Introduction

Présenter l’objectif du document et en résumé le contenu

Objectifs et méthodes

Présenter le développement du logiciel dans son ensemble.

Ce document décrit les tests fonctionnels, d’intégration et unitaires. Nous avons deux parties de test, la première a été effectué sur la base de données relationnelle, celle-ci étant faite sous PostgreSQL. La seconde partie concerne l’interface, celle-ci a été réaliser avec *Vue.js* et pour le lien entre la base de données et l’interface nous avons utilisé *Node.js* et *Express.*

Le but est d’avoir une interface accessible qui permet d’interroger la base de données pour observer les informations d’un individu à la fois.

Documents de référence

Lister tous les documents du projet servant à l’élaboration du présent document.

Pour l’élaboration de notre projet plusieurs documents ont été mis à notre disposition. La thèse de David Noël a permis de suivre un modèle multidimensionnel. Nous avons également eu plusieurs documents permettant de comprendre les données fournies par l’INED.

Concepts de base

Précise les concepts de base nécessaires a la compréhension du document.

Afin de comprendre sans difficulté le document, voici quelques définitions :

***Les trajectoires de vie :*** Représente le parcours d’un individu au niveau spatio-temporel.   
***Modèle multi point de vue :*** Plusieurs point de vue, ici nous avons familiale, professionnel, résidentiel et de voyage. Chacun peut être vu comme une dimension de la trajectoire.

***Frontend :*** Cette notion se reporte à tous les éléments qui sont visibles ou accessible depuis le site web ou l’application web.

***Backend :*** Cette notion se reporte à tous les éléments qui sont utile pour le développement, la base de données.

Test fonctionnels

Pour chaque scenario

Décrire les scenario (ainsi que leur enchainement) permettant de vérifier que l’application recouvrira bien le périmètre fonctionnel qui a été définit lors de la phase de spécification. Le périmètre fonctionnel est défini dans le cahier des charges (expression fonctionnelle du besoin)

Identification

Scénario 1 – Interrogation d’un individu

Description

Décrire le but du test, les caractéristiques de l’environnement de test et le principe de réalisation du test.

Contraintes

Décrire les contraintes liées à ce scénario : environnement de test particulier, installation particulière, intervention humaine spécifique, etc …

Indiquer les éléments de documentation faisant référence (spécifications, éléments de conception, etc …)

Dépendances

Lister et expliciter les tests à mener préalablement à la réalisation du scénario.

Procédure de test

Décrire les données en entrée, les résultats attendus, et les critères de validation

Test d’intégration

Pour chaque test d’intégration

Décrire les tests (ainsi que leur enchainement) permettant de vérifier que les différents modules, paquetages, … de l’application s’interfacent correctement. On distinguera les interfaces internes des interfaces externes à l’application.

On peut d’aider de :  
https://github.com/mkosareva/IA-Toby/blob/master/Documents/Plan\_de\_test\_Toby.pdf

## Validation de la base de données

Description

Le fichier d’origine des données possède environ 500 000 lignes et 14 colonnes d’informations. C’est une grosse quantité d’information et il est fréquent que la manipulation des données à l’intérieur d’un tel fichier entraine des pertes, endommagements ou autre. De plus, il faut faire attention à ne pas répéter les possibles informations redondantes dans la base de données.

But

L’objectif ici est de s’assurer de la concordance entre le fichier csv et la base de données finale. En effet, on ne veut pas perdre de données lors de la mise en forme de ces dernières.

Réalisation

La base de données produite à partir du fichier csv possède des tables au format similaire :

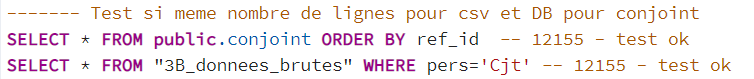
1. parent – conjoint – enfant
2. professionnal – residential – leisure
3. localisation
4. personne
5. familial

Ci-dessous est disponible la démarche de validation pour chaque format. On utilise la table ‘3B\_donnees\_brutes’ pour observer les données du fichier csv, car ils sont identiques.

1. On regarde le nombre de ligne dans le fichier csv avec l’indice correspondant, par exemple ‘Cjt’, il y en a 12155.
2. On regarde le nombre de ligne de la table correspondant à l’indice dans la base de données, c’est-à-dire ‘Conjoint’. Il y a 12155 lignes.
3. Le nombre de ligne correspond, le test est positif.

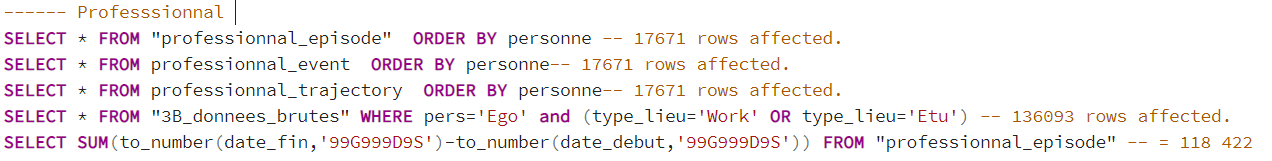
Voici ci-dessous les requêtes PostGreSQL.

**Table Conjoint :**



On vérifie qu’il y ait le même nombre de ligne dans la table Conjoint et dans la table « 3B\_donnees\_brutes ».

**Table Professionnel :**



1er : Vérifie qu’il y a autant d’évènements, d’épisode et donc de trajectoire.

2ème : Compte le nombre de ligne associé aux attributs récupérer pour les tables professionnelles.

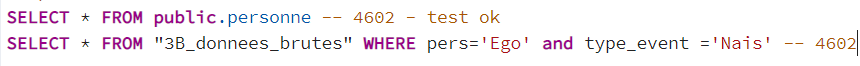
3ème: Compte le nombre de ligne qui correspondent a un même épisode. Car avant on a plusieurs lignes pour un épisode constant alors que dans la BDD on a plus qu’une seule ligne qui a une date de début et une date de fin.

4ème: On fait 136093 – 448422 = 17671. Donc on a bien gardé toutes les informations.

**Table Localisation :**

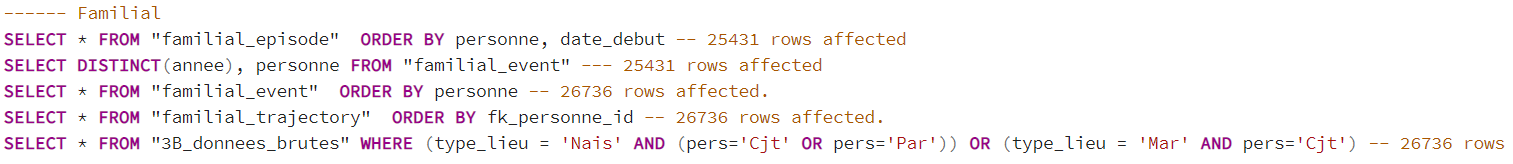
On vérifie qu’il y a le même nombre de ligne dans la table Localisation que de différente localisation dans la table « 3B\_donnees-brutes ».

**Table Personne :**



On vérifie qu’on a bien toutes les personnes présente dans « 3B\_donnees\_brutes ».

**Table Familial :**



1er : Vérifie que le nombre d’épisode est également au nombre d’évènement en enlevant ceux qui se déroule la même année. On a donc bien le même nombre.

2ème : On vérifie qu’évènement et trajectoire correspondent.

3ème: On vérifie que cela correspond à « 3B\_données\_brutes » ce qui est le cas.

Test unitaires

Pour chaque test unitaire

Décrire les tests (ainsi que leur enchainement) permettant de vérifier que le comportement de chaque fonction, méthode, etc… de l’application corresponde de manière satisfaisante à ce qui a été défini lors de la phase de conception détaillée.

Identification

A faire

Description

A faire

Contraintes

A faire

Dépendances

A faire

Procédure de test

A faire

Vérification de la documentation

Vérifier la concordance entre les différentes procédures décrites dans la documentation et leur mise en œuvre réelle.

Annexes

A faire

Glossaire

Définit l’ensemble des termes spécialisés du document

Références

Indique les références bibliographiques vers d’autres documents apportant des informations complémentaires

Index

Liste les mots-clés du document et où les trouver dans celui-ci