Spécifiation de conception de la base de données : Jalon 2

Louis-Vincent CAPELLI Alexandre THEISSE
Tom SARTORI Raphaël TURCOTTE

October 29, 2023

Contents

1 Introduction

Objet et portée du document

Ce document a pour but de documenter la création d'une API sur la base de données SSQA afin de répondre aux exigences formulées dans la spécifiation des exigences du modèle pour le jalon 2. Il s'adresse à toute personne qui pourrait avoir à travailler sur la base de données SSQA à l'avenir.

2 Présentation générale du résultat

L'ensemble de l'API est en PostgreSQL comme la base de données SSQA.

Nous avons décidé de créer les fonctions atomiques correspondant aux primitives fondamentales d'ÉMIR ainsi que les fonctions de production de rapport dans un nouveau schéma public nommé `"SSQA_PUB"` afin de permettre à l'utilisateur de les utiliser sans avoir à connaître le schéma privé.

3 Choix de conception

3.1 Primitives fondamentales

Description Implémenter les primitives fondamentales d'ÉMIR pour les tables Territoire, Station, Variable et Mesure.

Solution choisie Dans le cadre de cette requête, nous avons décidé de créer différentes fonctions et procédures pour chaque primitive fondamentale. En effet, cette division des primitives fondamentales EMIR en plusieurs fonctions et procédures permet de simplifier la compréhension de la base de données et de faciliter la maintenance de celle-ci. Cette décomposition des fonctions tient compte des différentes liens qui existent entre les tables.

Par exemple, la procédure d'insertion générale d'une mesure insère également une erreur de mesure si la valeur en argument est en dehors de l'intervalle de validité de la variable pour la norme précisée.

Nous avons ainsi créé une quantité arbitraire de fonctions notamment pour l'évaluation, avec, toujours pour la table Mesure, les fonctions suivantes :

- `mesure_eva_gen` sélection de toutes les mesures
- `mesure_eva_stat` sélection de toutes les mesures d'une station
- `mesure_eva_var` sélection de toutes les mesures d'une variable
- `mesure_eva_date` sélection de toutes les mesures d'une date
- `mesure_eva_stat_var` sélection de toutes les mesures d'une station et d'une variable

- `mesure_eva_stat_date` sélection de toutes les mesures d'une station et d'une date
- `mesure_eva_stat_var_date` sélection de **la** mesure d'une station, d'une variable et d'une date
- `mesure_eva_stat_var_after_date` sélection de toutes les mesures d'une station, d'une variable et ayant eu lieu après une date
- `mesure_eva_stat_var_between_date` sélection de toutes les mesures d'une station, d'une variable et ayant eu lieu entre deux dates

Enfin, nous avons activé la suppression en cascade pour l'ensemble des tables, afin que les procédures de suppression soient fonctionnelles et plus simples à implémenter sans avoir à gérer les contraintes de clés étrangères. (cf. `3.0.0_SSQA_Foreign-key-on-cascade.sql`)

Solution alternative Concernant le dernier point, nous aurions pu créer des procédures de suppression pour chaque table et gérer les contraintes de clés étrangères à la main, mais cela aurait été plus long à implémenter et nous n'y voyions pas d'avantage.

3.2 Requête 1

Description Créer une fonction permet à l'utilisateur d'effectuer la requête suivante : "Quel est l'IQA découlant des mesures prises par la station du quartier universitaire de la ville de Sherbrooke, le 2016-06-12 ?"

Solution choisie Les fonctions de calcul de l'IQA sont dans les fichiers utilitaires nommés 3.5.0_SSQA_IQA.sql et 3.5.1_SSQA_IQA_mauvaisIQA.sql.

3.3 Requête 2

Description Créer une fonction permet à l'utilisateur d'effectuer la requête suivante : "Quelles sont les stations du territoire de la ville de Sherbrooke ?"

Solution choisie La fonction prend en paramètre le nom du territoire et retourne les stations. Cette fonction en fonction du paramètre territoire va d'abord chercher le territoire dans la table Territoire. Ensuite, elle va chercher les stations qui ont le même id_territoire que le territoire trouvé. Elle retourne les stations trouvées.

3.4 Requête 3

Description Créer une fonction permet à l'utilisateur d'effectuer la requête suivante : "Quels sont les quartiers de la ville de Sherbrooke qui dépassent la norme canadienne de la qualité de l'air au moins n fois par année ?"

Solution choisie La fonction prend en paramètre le nom du territoire, le nombre de fois que la norme est dépassée et l'année.

3.5 Requête 4

Description Créer une fonction permet à l'utilisateur d'effectuer la requête suivante : "Dans un territoire donné, au cours de l'année 2021, quels sont les polluants qui ont dépassé la valeur de référence d'au moins 10% ?"

Solution choisie La fonction prend en paramètre le nom du territoire et l'année et retourne les polluants qui ont dépassé la valeur de référence d'au moins 10%.

3.6 Requête 5

Description Créer une fonction permet à l'utilisateur d'effectuer la requête suivante : "Quels sont les territoires ayant au moins trois fois par mois un mauvais IQA au cours des deux dernières années ?"

Solution choisie La fonction prend en paramètre le nombre de fois que la norme est dépassée, et le nombre d'années consécutive qui ont vu eu un mauvais IQA. Les fonctions qui vérifient si l'IQA est mauvais sont des les fichiers utilitaires cités plus haut.