Spécifiation de conception de la base de données : Jalon 1

Louis-Vincent CAPELLI Alexandre THEISSE
Tom SARTORI Raphaël TURCOTTE

October 11, 2023

Contents

1 Introduction

Objet et portée du document

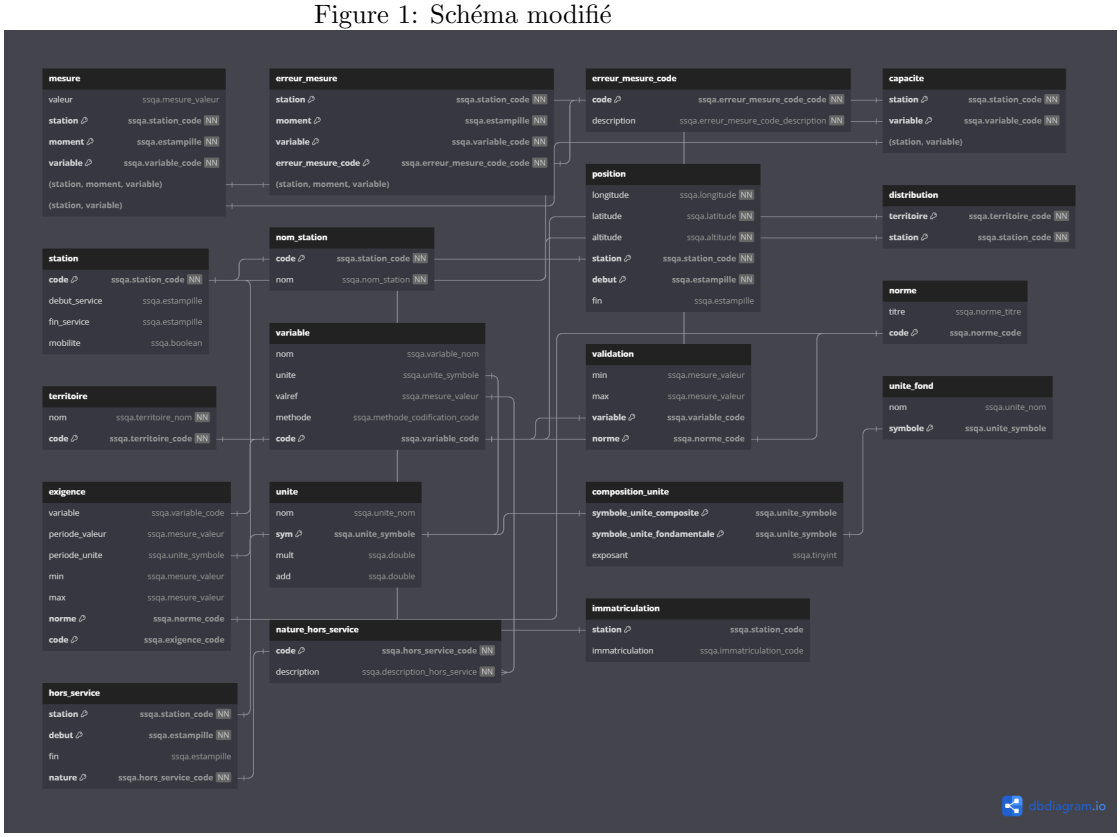
Ce document a pour but de décrire les modifications à apporter à la base de données SSQA afin de répondre aux exigences formulées dans la spécifiation des exigences du modèle. Il s'adresse à toute personne qui pourrait avoir à travailler sur la base de données SSQA à l'avenir.

2 Présentation générale du résultat

2.1 Schéma modifié

Voici ci-après le schéma modifié de la base de données SSQA. Il est en de nombreux points similaire au schéma préliminaire, mais comporte des modifications importantes, notamment au niveau de la représentation des unités et de la gestion de la validité des mesures. Ces modifications sont décrites plus en détail dans la suite du document, à la lumière des différentes exigences.

Figure 1: Schéma modifié



2.2 Normalisation

Le schéma modifié, après avoir été mis au point était déjà en forme normale de Boyce-Codd. Nous avons décidé de nous assurer qu'il était également en 5NF. Les relations de comprenant pas au moins un attribut non-clé sont (toutes les autres sont automatiquement en 5NF) :

- Capacité : qui est en 5NF car elle ne contient que deux attributs.
- Distribution : qui est en 5NF car elle ne contient que deux attributs.
- Erreur_mesure : si on la décompose la relation Erreur_mesure(station, moment, variable, erreur_mesure_code) en quatre sous-relations à trois éléments :
  - (station, moment, variable)
  - (station, moment, erreur_mesure_code)
  - (station, variable, erreur_mesure_code)
  - (moment, variable, erreur_mesure_code)

  on perd de l'information. Par exemple :
  - Station 1, 1/1/1970, concentration NO2 (peut être une erreur de manipulation)
  - Station 1, 1/1/1970, capteur surchauffe (peut être sur la concentration en CO2)
  - Station 1, concentration NO2, capteur surchauffe (peut être le 2/1/1970)
  - 1/1/1970, concentration NO2, capteur surchauffe (peut être dans la Station 2)

  On ne peut pas déduire de ces 4 tuples que "Station 1, 1/1/1970, concentration NO2, capteur surchauffe" donc on n'a pas à décomposer la table.

3 Choix de conception

3.1 R01.SI_mod

Description Représenter toutes les unités en fonction des unités fondamentales définies par le SI et de deux coefficients additif et multiplicatif.

Solution choisie Une unité est modélisée par un symbole et un nom. Sa définition est complétée par une valeur de multiplication et une valeur d'addition par rapport à l'unité fondamentale associée. Une table Unité_fondamentale permet de définir les unités fondamentales du SI (et d'en ajouter si nécessaire). Une table Composition_unité permet de définir les unités composées à partir des unités fondamentales, chaque entrée de cette table est composée d'une unité

fondamentale et d'un exposant et référence l'unité correspondante. Si l'unité est fondamentale, alors la valeur de multiplication est 1 et la valeur d'addition est 0 et elle n'est représentée que par une seule entrée dans la table Composition_unité.

Solution alternative Nous avions pensé à une autre solution pour représenter les unités comme le radian. Actuellement, le radian est ajouté comme unité fondamentale, mais nous aurions pu également ajouter toutes les unités du SI en double dans la table Unité_fondamentale, afin de pouvoir les utiliser avec deux exposants différents dans la table Composition_unité. Cela aurait permis de représenter le radian comme une unité composée de mètre et de mètre, et ainsi d'être plus cohérent avec la définition du SI. Cependant, cette solution aurait été plus lourde à mettre en place et moins facile à comprendre. Nous avons donc préféré la solution actuelle qui permet les mêmes fonctionnalités.

3.2 R02.SI_sym

Description Restreindre les symboles des unités grace aux règles du BIPM.

Solution choisie Nous n'avons pas implémenté cette exigence car nous n'avons pas trouvé de règles claires pour les symboles des unités. En supposant que l'utilisateur peut créer de nouvelles unités à volonté, il est impossible de prévoir la forme des symboles. Dans les exemples fournis avec le schéma préliminaire de l'EPP_V1, l'une des unités avait pour symbole µg/m3, mais on peut très bien imaginer des unités plus complexes avec des symboles comme µg*s2*A/m3.

3.3 R03.Validation_nom

Description Changer le nom de la table "Seuils" pour "Validation" et faire percoler les conséquences de ce changement dans le reste de la base de données.

Solution choisie Nous avons renommé la table Seuils en Validation et modifié les noms des contraintes correspondantes.

3.4 R04.Variable_contrainte

Description

- Vérifier que la valeur de référence de la variable est comprise dans l'intervalle de validité fixé par la norme associée.
- Vérifier que les min et max des exigences pour une variable sont compris dans l'intervalle de validité fixé par la norme associée pour cette variable.

Solution choisie Nous avons crée un trigger qui vérifie, avant l'insertion dans la table Validation, que la valeur de référence de la variable est comprise dans l'intervalle de validité. Si ce n'est pas le cas, l'insertion est refusée et l'erreur "L'intervalle de validation est invalide" est renvoyée.

Nous avons aussi crée un trigger qui vérifie, avant l'insertion dans la table Exigence, que les min et max sont compris dans l'intervalle de validité pour la variable et la norme concernées. Si ce n'est pas le cas, l'insertion est refusée et l'erreur "La valeur minimale de l'exigence est invalide" ou l'erreur "La valeur maximale de l'exigence est invalide" est renvoyée selon le cas.

3.5 R05.Méthode_codification

Description Les méthodes d'échantillonnage des variables sont en texte libre, ce qui laisse place à des erreurs de saisie qui pourraient résulter en la définition de plusieurs représentations pour une même méthode. Afin de mieux valider les données, les méthodes devraient être codifiées.

Solution choisie Nous avons considéré que les méthodes d'échantillonnage n'étaient pas censées évoluer dans le temps, et qu'il n'était donc pas nécessaire de pouvoir en ajouter de nouvelles. Nous avons donc décidé de changer le type de la colonne en un type enum, qui permet de restreindre les valeurs possibles à une liste prédéfinie.

Solution alternative Nous aurions pu également créer une table Méthode_échantillonnage qui aurait référencé les méthodes possibles, et qui aurait été référencée par la table Variable. Cela aurait permis de pouvoir ajouter de nouvelles méthodes d'échantillonnage à l'avenir.

3.6 R06.Station_service

Description Afin de permettre de valider les temps des mesures : ajouter les attributs de mise en exploitation et de fin d'exploitation de la station et maintenir une table des périodes d'entretien ou de non-disponibilité des stations.

Solution choisie Nous avons créé des tables Nature_hors_service et Hors_service qui permettent de définir les périodes d'indisponibilité des stations. La table Hors_service référence la table Nature_hors_service et la table Station. Elle contient également une date de début et une date de fin. La table Nature_hors_service contient un nom et une description et permet de définir les différentes raisons pour lesquelles une station peut être indisponible de manière évolutive au cours du temps.

Nous avons également ajouté une date de mise en exploitation et une date de fin d'exploitation à la table Station sous la forme de deux nouveaux attributs de type Estampille.

3.7 R07.Station_mobilité

Description Certaines stations sont mobiles, leurs coordonnées varient donc dans le temps. Les stations fixes peuvent aussi être déplacées à l'occasion. Ainsi il faudra modifier le schéma afin de pouvoir consigner l'évolution des coordonnées des stations.

Solution choisie Nous avons ajouté un attribut mobilite de type booléen à la table Station qui permet de savoir si une station est mobile ou non.

Nous avons créé une table Position qui référence la table Station et qui contient des dates de début et de fin, ainsi que des coordonnées. La table Position permet de définir les différentes positions qu'une station peut occuper au cours du temps.

Nous avons ainsi transféré les attributs latitude, longitude et altitude de la table Station vers la table Position en ajoutant les tuples correspondants aux anciens tuples de la table Station à la table Position avec une date de début égale à la date actuelle. Nous avons ensuite pu supprimer ces attributs de la table Station.

Nous avons également créé une table Immatriculation qui référence la table Station et permet de consigner les Immatriculations des stations mobiles comme expliqué dans l'EPP_V1.

3.8 R08.Validation_période

Description Vérifier que l'attribut periode_unite de la table Exigence est une unité de temps valide.

Solution choisie Nous avons créé un trigger qui vérifie, avant l'insertion dans la table Exigence, que periode_unite est bien une unité de temps en vérifiant que la table Composition_unité contient bien une seule entrée pour cette unité et que cette entrée référence bien l'unité fondamentale de temps (la seconde, 's').

Si ce n'est pas le cas, l'insertion est refusée et l'erreur "L'unité de la période n'est pas une unité de temps" est renvoyée.

3.9 R09.Station_nom_facultatif

Description Une station n'a pas forcément de nom, son emplacement est alors utilisé pour la désigner. Rendre le nom de la station facultatif dans la mesure ou elle n'est pas mobile.

Solution choisie Nous avons décidé de faire un projection jointure sur la table Station et de créer une table Nom_station qui référence la table Station et qui contient le nom de cette station. De cette manière, nom avons pu supprimer l'attribut nom de la table Station plutôt que de faire en sorte qu'il puisse être nul.

Nous avons donc transféré les tuples de la table Station vers la table Nom_station en ajoutant les tuples correspondants aux anciens tuples de la table Station à la table Nom_station et supprimé l'attribut nom de la table Station.

Nous avons égalemnt créé une fonction qui permet d'insérer une nouvelle station et qui ajoute ou non une entrée dans la table Nom_station selon si l'argument "mobilite" est vrai ou faux.

3.10 R10.Mesure_valeur_absente

Description Rendre la valeur d'une mesure facultative, et modifier la base de données afin de pouvoir conserver la cause de l'absence de mesure.

Solution choisie Nous avons créé deux tables Erreur_mesure_code et Erreur_mesure qui consigner les erreurs et leur cause sur les différentes mesures.

La table Erreur_mesure_code contient un code et une description et permet de définir les différentes causes possibles d'absence de mesure de manière évolutive au cours du temps.

La table Erreur_mesure référence la table Erreur_mesure_code et la table Mesure à travers les trois attributs station, moment et variable.

Nous avons créé des entrées dans la table Erreur_mesure pour chaque mesure antérieure où la valeur est absente en lui associant le code "-1" signifiant "Erreur inconnue".

Nous avons ensuite supprimé l'attribut valide de la table Mesure.

3.11 R11.Documentation

Description Afin d'assurer l'interprétation correcte des données, associer le texte complet de chaque prédicat ainsi que ses dépendances fonctionnelles à l'aide d'un commentaire inscrit au catalogue.

Solution choisie Nous avons respecté les conventions de nommage et commenté nos scripts SQL en y ajoutant les prédicats afin de faciliter la compréhension de la base de données.

Nous avons en revanche oublié d'ajouter les commentaires au catalogue dans la plupart des cas.