

Explication Modèle Conceptuel des Données Module Suivi Territoire

Compte rendu du débriefing le 11 juin 2018

Explication du MCD

Les tables : 15 tables.

- **t_modules** : stocker les modules qui sont propres à GeoNature.
- **cor_site_module** : la table de correspondance entre t_modules et t_base_sites. Permet de faire le lien entre un module et ses sites associés Plusieurs protocoles « gèrent » plusieurs sites différents (le site appliqué le module Suivi_ChauveSouris est différent de celui appliqué le module Suivi_Habitat par exemple), il faut donc une table de correspondance retrouver les sites de suivis d'un module
- **t_base_visits** : enregistrer les informations d'une visite : Sur quel site ? La date de visite ?
- **cor_visit_observer** : permettre de connaître le ou les observeurs d'une visite
- **t_role** : table des utilisateurs (observateurs ici)
- **t_base_sites** : la table de définition d'une site : son identifiant, son nom, etc.
- **cor_site_area** : la table de correspondance entre t_base_sites et l_areas. Permet de connaître les entités géographique présent sur un site. Utilisé ici pour identifier les mailles d'une ZP.
- **infos_site** : stocker les infos complémentaires d'un site. C'est une extension de t_base_sites pour mieux cerner sur quel ZP que l'on va travailler. Et notamment le taxon rattaché à une ZP.
- **TaxRef** : la table de gestion de taxons.
- **cor_visit_grids** : la table de correspondance entre une visite et une maille. Sur chaque maille, pour chaque visite, je note un taxon s'il est présent ou pas, ça permet de voir son évolution dans le temps. Chaque enregistrement de cette table correspond à une maille et une visite. Il faut donc la relier avec l_areas pour pouvoir accéder aux mailles correspondantes.
- **l_areas** : cette table stocke les données définissant une entité géographique (ça peut être un coeur de Parc, une réserve naturelle, une commune, une maille, etc). On peut savoir que cette entité géographique est de quel type grâce à l'attribut id_type (qui fait référence à la table bib_areas_types).
- **li_grids** : la table des mailles. (table fille de l_areas)
- **cor_visit_perturbation** : enregistrer les perturbations constatées lors d'une visite.
- **t_nomenclatures** : table Nomenclatures
- **bib_nomenclature_type** : la liste des types Nomenclatures.

Les contraintes de multiplicité : quelques exemples.

- La multiplicité 1-0..* entre t_base_sites et t_base_visits : chaque site peut avoir 0 à plusieurs visites. Cependant, chaque visite correspond seulement à un site.
- La multiplicité 1..*-1 : l_areas et cor_visit_grid: Une maille peut être visité plusieurs fois. Cependant, une visite d'une maille ne correspond à une seule commune.
- Et ainsi de suite.

Compte rendu du débriefing. Les choses à revoir, à modifier et à développer.

A modifier.

- Au début, les tables cor_site_app et t_base_visits sont reliées avec la table t_applications du schéma utilisateurs. Après avoir réfléchi, il faut lier ces 2 tables avec la table t_modules se trouvant dans gn_commons. Effectivement, nous voulons savoir sur cette zone de visite, quel protocole choisi et appliqué. Il est plus pertinent de les relier avec t_modules que t_application (qui regroupe toutes les applications, y compris UserHub, TaxHub, et qui gère les droits d'utilisateurs).

Par conséquent, la table cor_site_app devient cor_site_module, ce qui n'est pas encore existé et il faut la créer.

- On peut lier directement t_base_visits à t_modules sans avoir besoin d'une table correspondance. Car une visite ne correspond à un seul module.
- La table cor_visit_usage devient cor_visit_perturbation pour être compatible aux données existante de Geonature v1.
- Réaménager les tables pour optimiser la visibilité.

A développer.

- Tous ceux qui sont en vert sont à développer.
- Créer la table cor_site_module.
- Dans t_nomenclature, il manque le nomenclature des Perturbations. Il faut la créer.