

Le présent document est une note de travail rédigée lors de la revue du jalon IGE487_2025-3_TS_J1. Il comprend divers commentaires généraux applicables à plus d'un travail soumis.

Domaine du problème

- **Taux.**

- Le recours à une évaluation qualitative (même si elle correspond à une catégorisation approximative d'une mesure) entraîne une perte d'évolutivité non réversible.
- Si ce choix est maintenu, il doit être justifié et la contrainte de non-modifiabilité doit être intégrée au modèle.
- Il apparaît plus sage de passer à une mesure quantitative (au besoin avec une marge d'erreur propre à chaque mesure), sujette à une catégorisation ultérieure au moment de l'analyse.

- **Lieux.**

- À terme, les lieux devront intégrer les concepts de site et de zone, en raison de la distribution de BD indépendantes sur plusieurs sites. Ceci est essentiel puisque ces BD qui devront être fédérées ultérieurement et fort probablement à plusieurs reprises.
- Introduction d'un modèle hiérarchisé de désignation de lieu : site.zone.placette.parcelle.
- Systématisation grâce à des clés relatives.

- **Plants.**

- Les plants ont une identité propre indépendante du lieu où ils sont observés. Imaginez, par exemple, qu'il y ait un glissement de terrain - un même plant pourrait donc changer de lieu. Même si cette situation est rare (mais pas impossible), et même si les parties prenantes décident de ne pas l'intégrer dans leur modèle écologique, cette situation permet d'illustrer que l'identité du plan est indépendante de celle du lieu.
- Mettre en oeuvre l'identification absolue et l'emplacement relatif des plants.

- **Unité_de_mesure, Type_de_mesure.**

- L'introduction de ces concepts essentiels est-elle pertinente ? Deux solutions se démarquent :
 1. imposer un système de poids et mesure uniforme dans tout le modèle et prescrire l'unité de mesure appropriée dans chacun des prédicats concernés ;
 2. modéliser les unités de mesure, inclure un mécanisme de définition de types de mesure et prévoir un attribut permettant de le spécifier (en plus de la valeur) pour chacune de mesure.
- D'un point de vue informatique, les deux solutions sont adéquates.
- D'un point de vue pratique,
 1. la première est la plus simple à réaliser et la plus simple à utiliser ; toutes les requêtes, en particulier les requêtes analytiques, pourront être programmées et utilisées sans y intégrer les différentes conversions requises. Les conversions, si besoin est, sont faites au niveau des interfaces personnes-machines (comme le format des dates, la couleur de

la police d’affichage, etc.).

2. La seconde garde la trace de l’unité de mesure d’origine et permet de faire des conversions différenciées pour chacune des requêtes, dont les requêtes analytiques. Cette différenciation peut être un atout ou une plaie du point de vue de l’interprétation des résultats. Il est vraisemblable qu’il faudra prévoir au niveau des interfaces personnes-machines un mécanisme de conversion à l’affiche, comme pour la première solution.

- Le choix appartient aux parties prenantes, même s’il est possible de leur faire une recommandation !

Modélisation

- **M01.**

- Sous-utilisation des types spécifiques pouvant entraîner des failles de vérification (oubli, incohérence), une lourdeur notationnelle, une complexité non justifiée et des lacunes documentaires.
- Sur-utilisation de types génériques non contraints par définition possiblement compensée par l’utilisation de clés référentielles.

- **M02.**

- Dans les diagrammes conceptuels (Elmasri, Merise ou autre), ne pas inverser les participations (comme en UML), ne serait-ce que pour permettre la définition cohérente des associations de degré supérieur à 2.

- **M03.**

- Ne pas modéliser les entités faibles (EA) ou les dépendances fonctionnelles (Merise) à l’aide de clés internes (artificielles), ceci complexifie le modèle conceptuel (donc le diagramme et la communication avec les parties prenantes) et sont susceptibles de les imposer inutilement dans le modèle logique.

- **M04.**

- Au niveau du modèle conceptuel, le Taux doit être considéré comme un type et utilisé comme tel.
- Sa représentation à l’aide de tables est une décision qui relève de la conception, voire même de la programmation (au même titre que toute autre table de codage).

Conception

- **C01.**

- Définition d’une politique de gestion des données manquantes.
- Application au modèle logique et à l’ELT.

- **C02.**

- Dissocier systématiquement les tables d’alimentation utilisées comme zones de chargement lors de l’alimentation, des tables composant le modèle logique.

- Les premières sont normalement dépourvues de contraintes afin de permettre un chargement complet. Leur structure vise à simplifier le chargement des sources (indépendamment du modèle) ; elle tend souvent à refléter celle des sources.
- Les secondes sont contraintes au plus juste afin de garantir la cohérence, la validité et l'efficacité du modèle.

Programmation

• P01.

- Motiver tout recours aux spécificités dialectales, car la réduction de la transportabilité du code peut avoir des effets négatifs à long terme.
- Une telle décision doit donc être explicite et justifiée.
 - syntaxe des expressions rationnelles (*regular expression*) ;
 - opérateur :: plutôt que la fonction standard CAST
 - etc.

• P02.

- Ne pas reprogrammer les vérifications découlant de la définition des types.
- Utiliser CAST.
- Par exemple, pour les dates.

• P03.

- La rédaction et l'inclusion systématique des commentaires sont impératives afin de soutenir l'utilisation et l'évolutivité du modèle et de la BD.
- L'emploi de l'instruction COMMENT est encouragée afin de conserver les commentaires dans le catalogue de la base de données.

• P04.

- Une trace d'exécution des tests (lisible et bien présentée) doit être obtenue grâce à un script réutilisable (psql, bash, zsh, etc.).
- Ce script doit faire partie de la livraison.
- La trace d'exécution elle-même prend la forme d'un fichier de type texte (dont le nom comprend minimalement celui du script et le moment d'exécution ; dont le suffixe est typiquement .txt ou .log)
- Une trace d'exécution de chacun des scripts doit faire partie de la livraison.

• P05.

- Type ENUM ou table de code ?
- La décision repose sur l'évolutivité anticipée du type.
- Changer la définition d'un type ENUM signifie faire une commande ALTER pouvant avoir un impact sur la disponibilité à court terme de la BD. Il est aussi normalement beaucoup plus lourd de modifier les requêtes, fonctions et procédures pour refléter le changement de définition. Normalement, ceci doit être fait par les responsables de la DB à l'intérieur d'un

processus de développement suivi d'un déploiement.

- Changer la valeur d'une table de code peut être rendu disponible à l'ensemble des utilisateurs, comme celui de toute autre table.
- Les deux permettent la même qualité de validation des valeurs légitimes.

- **ÉMIR.**

- Dans un ÉMIR, tous les attributs non-clé de toutes les entités élémentaires (non dépendantes fonctionnellement d'une autre) doivent a priori pouvoir être modifiés, ce qui engendre généralement $n+1$ routine où n est le nombre de tels attributs.
- La routine supplémentaire permet de changer tous ces attributs en un seul appel (elle n'est donc pas obligatoire, mais fort utile en pratique).

Documentation

- **D01.**

- Appliquer les règles de styles relativement aux essais techniques.
- Phrases simples strictement coordonnées.
- Pas de figures de style.

- **D02.**

- Procéder aux vérifications et revues orthographiques, syntaxiques et stylistiques.

- **D03.**

- L'emploi d'outils de production documentaire afin de faciliter la rédaction des documents de conception et le maintien de la cohérence entre la documentation et le code est encouragé.
- Ceci permet d'éviter la redondance effective entre le document et le code en automatisant l'extraction des informations pertinentes du code et leur inclusion mise en forme adéquatement dans le document de conception.
- À défaut, la redondance entrainera des problèmes de vérification, de validation et d'évolution.
- Exemples de tels outils : SchemaSpy, SchemaCrawler, DataEdo, etc.

- **D04.**

- Le type de présentation sur fond sombre (noir) convient bien aux jeux vidéos lorsqu'on cherche à confondre le joueur et lui cacher de l'information.
- Il est fortement déconseillé dans le contexte d'un essai scientifique ou technique lorsqu'on cherche au contraire à être explicite, transparent et exhaustif tout en facilitant le travail du lecteur.