3eme partie Yasmine

# Agrégation des données:

## Préliminaires :

On a utilisé 2 tables *diag\_finale* et *anapath\_finale* qui contiennent le *newid* du patient avec la date de détection ; c’est la *datprel* pour l’anapath et la *edat* pour les diagnostics sans anapath.

On a ajouté avec R une colonne *etat\_diagnostic* qui nous renseigne la force du diagnostic:

Si c’est un diagnostic base sur une anapath on a mis “confirmé” si c’est sans anapath on a mis “probable”

Après vérification des tables on les a importés sur une base qu’on a appelé *cancer\_final*

Dans laquelle on a créé 2 procédures:

* Une procédure pour créer une vue des diagnostics sans anapath pour un patient donné (avec une jointure externe)
* Une procédure pour créer une vue des diagnostics avec anapath
* Une procédure pour créer une vue qui contient l’union des 2 tables : *table union*. Cette vue va nous servir pour remplir la table cancer.

## Création de la table cancer :

Algorithme utilisé en se basant sur les règles de comptage des cancers :

|  |
| --- |
| Pour chaque patient : |
| * Si cancer systémique (group\_morpho\_iacr 15 ) ou du système hematopoietique (groupes 8-14)  => 1 seul cancer (table cancer) |
| * Sinon : |
| * + Si même morpho |
| * + - Si même groupe topo => 1 seul cancer (table test) |
| * + - * Si diagnostiqué même jour on prend les codes dans table1 (remplacer un code) ( table test1) |
| * + - * Sinon On prend le code topo du 1er diagnostic (remplacer un code) (table test2) |
| * + - Sinon on prend tout (table test3) |
| * + Si différente morpho : |
| * + - Si même groupe morpho |
| * + - * + Si même topo |
| 1 seul cancer avec le plus élevé code CIM03morpho (table test5) |
| * + - * + Sinon on prend tout (table test6) |
| * + - Si différents groupes morpho on prend tout (table test4) |
|  |
| Ajouter test1,2,3,4,5,6 dans table cancer (insert into select distinct) |
| Remplacer le contenu de la table cancer par (distinct\* from cancer) à l’aide d’une table intermédiaire ( table test7) |

On a implémenté cet algorithme à l’aide de PHP et phpMyAdmin :

On a commencé par parcourir la table et extraire chaque newid avec une requête select distinct id

Et pour chaque newid on va continuer à tester.

Pour les cancers systémiques une simple requête select distinct\* a permis d’insérer les lignes correspondantes sur la table cancer.

Pour le reste on a utilisé une fonction *recupmorpho* qui va extraire avec le même principe les codes morpho pour chaque patient.

Pour chaque code morpho récupéré on a appliqué la fonction morpho. Celle-ci va teste si on a le même code morpho pour un patient donné ou pas en faisant une requête *select count* et voir le résultat.

Si le résultat est >1 alors le code se répète et donc on va continuer à tester sinon il est unique et passe au cas « si morpho différente »

Le même principe est appliqué à chaque fois pour teste si on a le même groupe topo ,même jour de diagnostic etc.

Pour le cas ou 2 diagnostics sont faits le même jour , on a inséré ces lignes sur une table test1 et à chaque fois on modifie suivant les règles de la table 1 dans le document « règles de comptage des cancers »

Pour prendre le code topo du 1er diagnostic il suffit de faire une requête qui fait *un group by having date=max(date).*

Même principe pour avoir le code cimo3 morpho le plus élevé.

Pour vérifier la cohérence des résultats on a utilisé des tables intermédiaires (*test1-6*), qu’on a regroupé après et inséré dans la table cancer.

## Finalisation de la table cancer :

Pour faire cela on a utilisé la page *finalisation\_cancer.php*

Cette page exécute un code qui permet de réunir toutes les tables tests dans la table cancer et puis exécute un code qui permet de modifier cette table selon ces règles :

|  |
| --- |
| Pour chaque patient : |
| * Si même topo : |
| * + - Si groupe morpho not in(5-14-17) => on a une morpho spécifique |
| * + - * Effacer les 4-14 et 17 |
| * + - Si groupe morpho 5 |
| * + - * Effacer 15 et 17 |
| * + - Si groupe morpho14 |
| * + - * Effacer 17 |
|  |

Ajouter un id\_cancer en auto\_increment pour exploiter la table.

## Exploitation de la table cancer :

On a lié cette table *cancer* aux tables *tab\_patient* et table *transcodage* et on a crée un site web pour visualiser tous les résultats.

Pour les cancers, on a choisi de permettre à l’utilisateur d’afficher tous les cancers et aussi de faire des recherches selon des critères bien définis.

Les résultats sont affichés sur le site et peuvent être exportés sous format csv pour permettre de des analyser.

## Limites :

Lors de l’exportation du résultat en csv les caractères accentués ne sont pas pris en charge.

## Forces :

Sécurité optimisée en frontend avec les patterns et en backend avec la vérification des inputs à l’aide de htmlspecialchars qui n’autorise pas les injections SQL dans la base de données.

Sécurité de tout le site est assurée par un mot de passe et un nom d’utilisateur qui doivent être valides.

Pour plus de sécurité on a ajouté une captcha, ce qui permet d’éviter les attaques par brute-force.