Cas Dane - proposition de corrigé

Mission 1 - Nouvelles statistiques

D1.1 Analyse de la demande
D2.3 Gestion des problèmes et des changements
D4.1 Conception et réalisation d'une solution applicative
• Conception ou adaptation d'une base de données

1.1 - Écrire les requêtes *SQL* qui répondent aux besoins exprimés par la Dane.

1. **Liste des noms des applications répertoriées dans la catégorie d'usages "Parcours pédagogique".**

```
SELECT nom as "Applications du Parcours Pedagogique"
FROM ApplicationENT
JOIN CategorieUsage ON idCategorieUsage = CategorieUsage.id
WHERE libelle = 'Parcours pédagogique'
```

Remarque : l'alias sur le nom n'est pas éxigé

2. **Afficher pour l'établissement 9940001A (lycée Vanille Bourbon), le minimum, le maximum et la moyenne du nombre de connexions pour l'année scolaire 2015-2016 (du 1er septembre 2015 au 5 juillet 2016).**

```
SELECT MIN(nbCnx) as "nombre minimal de connexions", MAX(nbCnx) as "nombre maximum de connexions", AVG(nbCnx) as "moyenne mensuelle de connexions"
FROM HistoCnxEtablissement
WHERE rneEtablissement = '9940001A'
AND dateObservation BETWEEN '01/09/2015' AND '05/07/2016'
```

3. **Liste des établissements (libellé et type) n'ayant aucune application répertoriée dans la catégorie d'usages "Travail collaboratif".**

```
SELECT libelle, type
FROM Etablissement
WHERE rne NOT IN (SELECT DISTINCT rneEtablissement
                  FROM AppEtabActivees
                  JOIN ApplicationENT ap ON idApplicationENT = ap.id
                  JOIN CategorieUsage c ON ap.idCategorieUsage = c.id
                  WHERE libelle = 'Travail collaboratif' )
```

Autre solution:

```
SELECT libelle, type
FROM Etablissement e
WHERE NOT EXIST(
        SELECT *
         FROM AppEtabActivees a
        WHERE e.rne = a.rneEtablissement
        AND idApplication IN (
                SELECT id
                FROM ApplicationENT ap
                JOIN CategorieUsage c ON ap.idCategorieUsage = c.id
                WHERE libelle = 'Travail collaboratif')
                )
```

D2.3 Gestion des problèmes et des changements
D4.1 Conception et réalisation d'une solution applicative
• Développement, utilisation ou adaptation de composants logiciels

2.1- Expliquer l'intérêt de ce nouveau graphique pour le tableau de bord.

- Dans le premier graphique, le fort pourcentage des usages 'vie scolaire' « écrase » tous les autres usages. On ne peut plus vraiment les différencier. Ce nouveau graphique est un camembert dédié. Il permet donc de comparer les usages autres que la vie scolaire les uns par rapport aux autres.
- Il est important de disposer d'un graphique uniquement concentré sur les usages du numérique pédagogique, cœur de métier de la Dane.

2.2- Indiquer sur votre copie, les lignes de codes nécessaires pour prendre en compte la note rédigée par M. Vans.

→Modification du script **report.js**

Il faut ajouter la fonction **cnx_usages_horsVS(rne, dateDebut, dateFin)**. Cette fonction sera quasi identique à **cnx_usages(rne, dateDebut, dateFin)**. Elle définira les propriétés du nouveau graphique et enverra au script **report_switch.php** les paramètres nécessaires pour appeler la nouvelle méthode de la classe **ReportEtab** qui retournera les données attendues.

```
function cnx_usages_horsVS(rne, dateDebut, dateFin) {

        var optionsCnxUsages = {};
        optionsCnxUsages.chart = {
                renderTo: 'graph_cnx_usages_horsVS', backgroundColor: '#E6F8E0',
            borderColor: '#BDBDBD', borderWidth: 3
         };

        optionsCnxUsages.title = {
          text: 'Répartition des visites par catégorie d\'usages (hors VS)',  style:
{fontWeight: 'bold'}
        };

        $.getJSON('report_switch.php?action=EtabRecapUsagesHorsVS&rne=' +
rne+'&dateDebut='+dateDebut+'&dateFin='+dateFin, function(dataCnxUsages) {

          optionsCnxUsages.data = {total : 0, stats : []};
          $.each(dataCnxUsages, function(i, e){
             optionCnxUsages.data.total += e.nb_cnx;
             optionCnxUsages.data.stats.push({usage : e.usage, nb_cnx : e.nb_cnx});
          });
          chartCnxUsages = new Highcharts.Chart(optionsCnxUsages);
         });
        }
```

→Modification du script ***report_switch.php***

Ce script doit permettre de lancer, à partir des informations fournies lors de son appel (paramètres ***action, rne, dateDebut et dateFin***) la nouvelle méthode **j*sonRecapUsagesHorsVS()*** de la classe ***ReportEtab*** permettant de retourner les données attendues.

On ajoutera la ligne suivante dans le *switch($action)* :

```
case 'EtabRecapUsagesHorsVS':
    echo $report->jsonRecapUsagesHorsVS();break;
```

→Modification du script ***Report.class.php***

Il faut ici ajouter la méthode ***jsonRecapUsagesHorsVS()***. Cette méthode sera similaire à ***jsonRecapUsages()***. Seule la définition de la requête changera avec l'ajout d'une restriction concernant le type d'usage « Vie Scolaire ».

```
class ReportEtab {
…
public function jsonRecapUsagesHorsVS()  {
      $sql = 'SELECT libelle AS usage, SUM(nbCnx) AS nb_cnx'.
         ' FROM HistoUsageEtablissement INNER JOIN CategorieUsage '.
        ' ON id = idCategorieUsage'.
        ' WHERE rneEtablissement = :rne '.
        ' AND dateObservation BETWEEN :dateDebut  AND :dateFin '.
→       ' AND idCategorieUsage <> 108 '.
        ' GROUP BY libelle';
      $res = DbCnx::getCnx()->prepare($sql);
      $res->bindValue(':rne', $this->rne, PDO::PARAM_STR);
      $res->bindValue(':dateDebut', $this->dateDebObs, PDO::PARAM_STR);
      $res->bindValue(':dateFin', $this->dateFinObs, PDO::PARAM_STR);
      $res->execute();
      $lines = $res->fetchAll(PDO::FETCH_ASSOC);
      $res->closeCursor();
      return json_encode($lines);
   }
```

2.3- Quelle solution pouvez-vous proposer à M. Vans afin d'éviter d'écrire la fonction *cnx_usages_horsVS* (sans mettre en œuvre la solution).

Une solution basée sur la méthode initiale ***cnx_usages*** pourvue d'un paramètre précisant s'il s'agit d'indicateurs avec *VS* ou hors *VS* serait plus efficace en cas de maintenance ou de modification du traitement.

D2.3 Gestion des problèmes et des changements
D4.1 Conception et réalisation d'une solution applicative
• Développement, utilisation ou adaptation de composants logiciels

3.1- Écrire la méthode *getNbEtablissements()* de la classe *Audiences*, qui retourne le nombre d'établissements chargés dans les dictionnaires (la méthode pourra se baser indifféremment sur le dictionnaire *$lesConnexions* ou *$lesApplications*)

```
public function getNbEtablissements() {

        return $this->lesConnexions->count();
}
```

3.2- Écrire la méthode *syncEtablissements()* de la classe *Synchronisation*, qui se charge de déceler les nouveaux établissements qui doivent être ajoutés en comparant les deux sources de données. En présence d'un nouvel établissement, la mise à jour de la base se fera à l'aide de la méthode *AudiencesBdd::insertEtablissement()*.

La comparaison des deux sources de données doit être réalisée avant de parcourir le dictionnaire.

```
public function syncEtablissements (){
        if($this->lAudienceCsv->getNbEtablissement() >
                                        $this->lAudienceBdd->getNbEtablissement()){

                foreach($this-> lAudienceCsv->getLesConnexions() as $cle => $valeur){

                        if( ! $this->lAudienceBdd->getLesConnexions()->offsetExist ($cle)){
                                $unEtab = cle + ";" + $valeur[0] + ";" + $valeur[1];
                                $this-> lAudienceBdd->insertEtablissement($unEtab);
                        }
                }
        }
}
```

3.3- Expliquer l'origine de cette erreur et donner le correctif à apporter au code.

L'erreur retournée provient du système de gestion de la base de données qui a détecté le non-respect d'une contrainte d'intégrité référentielle. Cette erreur est provoquée lors de la synchronisation, au moment où la méthode *syncAll()* appelle la méthode *syncConnexions()*. En effet, le nouvel établissement du jeu d'essai n'a pas encore été ajouté dans la base et l'on cherche à mettre à jour ses différents nombres de connexions.

Il faut lancer la synchronisation des établissements <u>avant</u> d'effectuer les mises à jour. La méthode *syncAll()* devient :

```
public function syncAll() {
        $this->syncEtablissements();
        $this->syncConnexions();
        $this->syncApplications();
}
```

D1.1 Analyse de la demande
D2.3 Gestion des problèmes et des changements
D4.1 Conception et réalisation d'une solution applicative
• Développement, utilisation ou adaptation de composants logiciels
• Conception ou adaptation d'une base de données

4.1- Expliquer les inconvénients de la gestion papier des enquêtes tant pour M. Santes et la secrétaire de la Dane que pour les référents.

.

<u>Pour les référents</u> : s'assurer d'avoir à disposition les formulaires papier, les remplir manuellement avec des risques d'erreur de saisie, ratures, les transmettre à la DANE, risques de perte, délais pour ce processus,….

<u>Pour la Dane</u> : la secrétaire devra vérifier la bonne réception de tous les formulaires, les saisir sur un tableur, risques d'erreur de saisie, coûts financiers (papier, impressions), pas d'éco-responsabilité…
Pour M. Santes : accès aux statistiques compliquées, fiabilité des données, disponibilité des données pour traitement,

4.2- Expliquer en quoi l'utilisation de la nouvelle fonctionnalité en ligne va changer les habitudes des référents et celles de la Dane. Préconiser les solutions permettant de gérer ce changement.

<u>Pour les référents et pour la Dane</u> : frein au changement, temps d'adaptation, prise en main => mauvaise utilisation, erreur de saisie pour le référent. En revanche, la secrétaire n'aura plus à réceptionner les formulaires et à en saisir le contenu dans des feuilles de calcul, économie possible de papier et d'impressions.

<u>Solutions</u> : information, formation, participation des utilisateurs, assistance aux utilisateurs.

4.3- Donner des recommandations sur l'environnement de développement à mettre en place, à la fois sur les plans logiciels et matériels, pour la programmation de la nouvelle fonctionnalité.

Plan matériel :

- **serveur de données indépendant du SI (éventuellement virtuel) ;**
- **poste client de développement.**

Plan logiciel :

- **copie de la partie de la BD concernée sur ce serveur ;**
- **IDE de développement sur un poste client.**

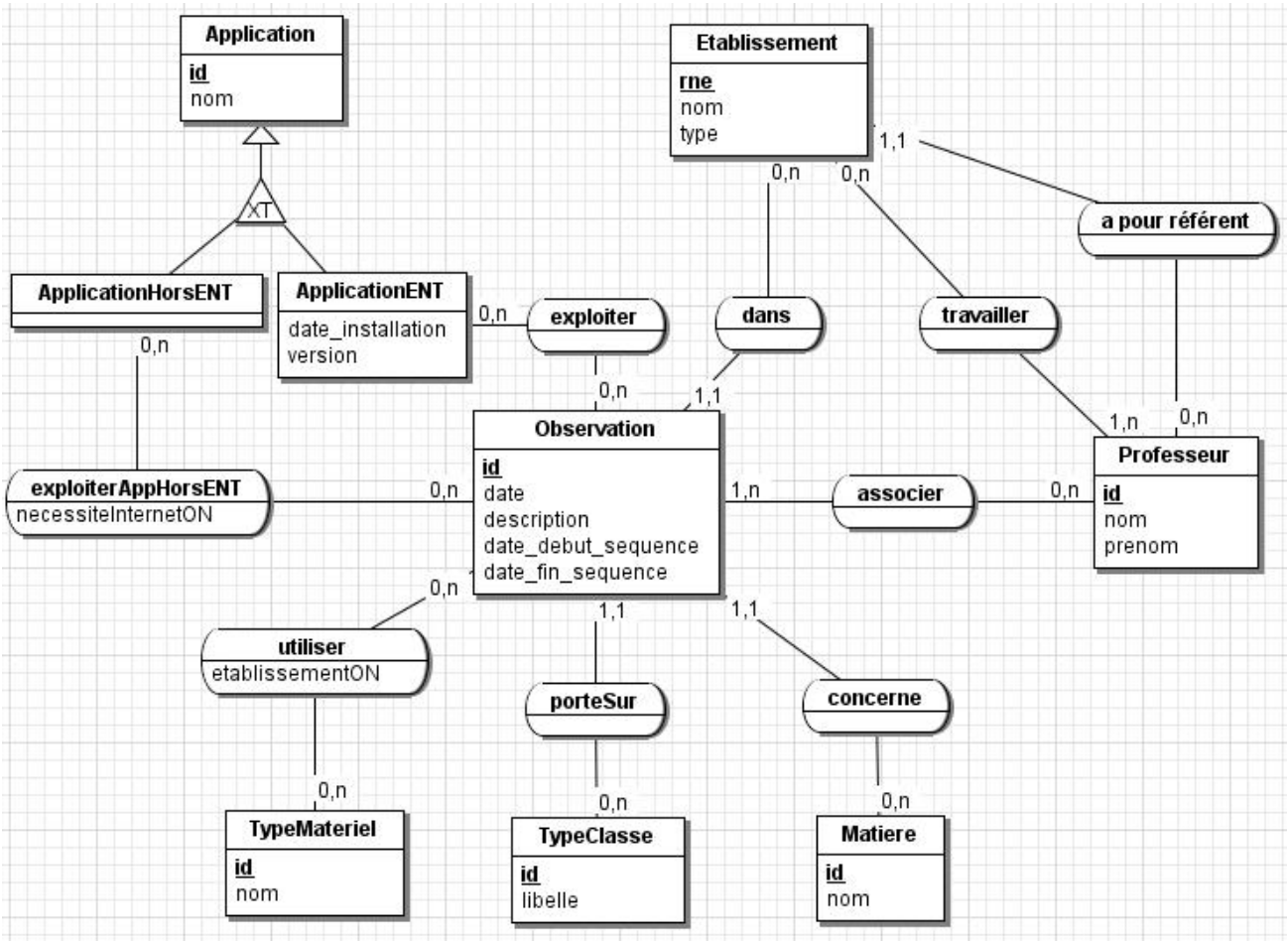
Évolution de la structure de la base de donnés

4.4- Proposer pour chacune des zones 1, 4, 8 et 10 le composant graphique adapté (zone de saisie, liste déroulante, etc...) pour limiter les erreurs de saisie.

Zone 1 : Liste déroulante (comboBox)
Zone 4 : Composant date (Calendrier)
Zone 8 : Zone de texte multi-lignes
Zone 10 : Case à cocher (checkBox)

4.5- A partir de la maquette et du schéma conceptuel existant (*document 12*), proposer une modélisation de la base de données qui prenne en compte les nouveaux besoins exprimés.

Version modèle entité association



Version diagramme de classe