Suivi des formations

(projet élève basé sur le contexte de niveau 2)

Description du thème

Propriétés	Description
Intitulé long	Suivi des formations des agents par leurs chefs de service
Formation concernée	Classe terminale de la série Sciences et technologies de la gestion (STG) Spécialité « Gestion des systèmes d'information » (GSI)
Matière	Gestion des systèmes d'information
Présentation	Basé sur le contexte 2, cette évolution propose de développer une application utilisée par les chefs de service lors d'entretiens d'évaluation annuels afin de suivre les formations suivies par les agents
Notions	A1.3 évolution du système d'information B1.1 définition, interrogation et mise à jour des données C1.2 formalisation des besoins C2 adaptation de l'application
Outils	MySQL, PhpMyAdmin, Access, MySQL ODBC 3.51
Mots-clés	contexte, évolution, projet examen, objectif, contrainte, cas d'utilisation, maquettage, programmation
Durée	64 h = 16h/élève * quatre élèves
Auteur(es)	Jean-Philippe Pujol avec les conseils de Christine Gaubert-Macon et Eric Vaccari
Lien(s)	contexte 2 : http://www.reseaucerta.org/cotecours/cotecours.php?num=362 actualisation des dates : http://www.reseaucerta.org/outils/outils.php?num=371
Version	1.0
Date de publication	octobre 2006

Les élèves de terminale GSI présentent au baccalauréat une épreuve de spécialité dont la partie pratique est dénommée « projet »
Dans une activité de groupe d'une durée de 16h, trois à quatre élèves participent à sa réalisation.

Le professeur de spécialité, dans le cadre de l'horaire de travaux dirigés, conseille et guide les élèves.

Ce document présente un contexte de projet et propose une piste de conduite pour le professeur et les élèves.

Des éléments techniques de corrigé sont proposés dans le répertoire « élémentsCorrigé ».

Contexte

Ce projet est basé sur le contexte 2 « Gestion des formations » publié sur le site du Certa (http://www.reseaucerta.org/cotecours/cotecours.php?num=362) auquel le lecteur voudra bien se référer pour appréhender les détails du système d'information proposé.

Résumé du contexte 2 :
Des formations internes à une organisation sont proposées à des agents ; les employés du service compétent gèrent les inscriptions, les envois de convocation, les saisies des présences puis les remboursements des frais d'hébergements.
Actuellement, une application de bureau mettant en œuvre Access assure ces traitements avec des données hébergées dans une base MySQL.

Évolution du système d'information

Les chefs de services réalisent un bilan annuel avec les agents dont ils ont la responsabilité. Ce bilan permet, outre une appréciation du travail réalisé, d'envisager des évolutions de carrière.

À l'issue de cet entretien de bilan annuel, les chefs de service sont parfois amenés à prescrire une ou plusieurs actions de formation à un agent. Cette formation peut aussi bien servir à consolider les compétences face à une évolution du poste occupé qu'à prétendre à une promotion. Les éléments importants du bilan sont consignés sur une fiche individuelle.

Les agents ont la maîtrise totale de leur plan de formation. Ils peuvent s'inscrire aux sessions prescrites par leur chef de service ou refuser cette prescription. Ils peuvent aussi s'inscrire à d'autres sessions qui leur semblent intéressantes.

Les chefs de service souhaitent mémoriser les actions prescrites lors des entretiens annuels puis visualiser l'année suivante les sessions qui ont été suivies réellement par les agents.

Démarche

Définition des besoins

Chaque chef de service doit pouvoir disposer d'une application lui permettant de réaliser le suivi de ses prescriptions d'actions de formation, de connaître les formations auxquelles se sont inscrits les agents ainsi que celles qu'ils ont suivies.

Définition des objectifs

Les fonctionnalités de l'application peuvent être résumées :
- saisie d'une prescription pour un agent ;
- consultation des formations prescrites ou demandées par un agent avec indication du suivi effectif.

Contraintes imposées

L'application sera mise à disposition des différents chefs de service. Ils ne doivent pouvoir gérer que les agents dont ils ont la responsabilité.
Chaque agent est identifié par un code connu de son chef de service. Un code est composé de 7 caractères numériques suivis d'un caractère alphabétique correspondant à un contrôle modulo 23.
La liste intégrale des agents ne sera pas directement accessible et ils devront saisir le code d'un agent pour accéder à ses informations. Un contrôle de ce code assurera une sécurité suffisante.
Le système mémorise :
- les sessions et inscriptions de l'année en cours ;
- les prescriptions de l'année passée et celles de l'année en cours.

L'application sera développée à l'aide d'Access. Elle devra se connecter à la base de données existante MySQL via un médiateur ODBC.

Étapes de la réalisation

Le projet proposé peut être décomposé en différentes étapes

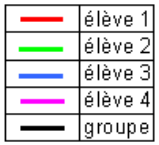
étape	travail du groupe
Production d'un document : - synthétisant l'activité de l'organisation ; - présentant et justifiant le besoin d'évolution du système d'information ; - définissant les fonctionnalités attendues ; - présentant les contraintes.	Chaque élève rédige une partie du document. Ensemble, ils agrègent et corrigent les éléments puis réalisent la synthèse.
Réalisation des cas d'utilisation	Pour satisfaire les deux fonctionnalités attendues les élèves réalisent par couple un cas d'utilisation ; chaque couple valide ensuite celui de l'autre.
Évolution de la base de données	À partir de la base de données existante, les élèves décident de l'évolution qu'elle doit subir pour satisfaire les fonctionnalités attendues. Ils réalisent ensuite la modification de structure de la base.
Gestion de la sécurité	L'accès à la base doit être réservé aux chefs de service. Les élèves créent un nouvel utilisateur disposant d'autorisations spécifiques sur certains objets de la base.
Initialisation des données	Les élèves doivent enregistrent dans la base de données un certain nombre de prescriptions de formations à partir de données statistiques. Ils déterminent les valeurs des données à insérer puis créent des requêtes SQL d'insertion.
Connexion à la base	La connexion à la base doit permettre l'exploitation des données par Access. Les élèves construisent une connexion ODBC utilisant le médiateur MySQL ODBC avec les paramètres adéquats.
Maquettage de l'application	Les élèves construisent les maquettes des interfaces.
Développement de l'application	Les élèves identifient les fonctions communes aux différents cas d'utilisation. Ils développent : - les fonctions communes placées dans un module ; - les procédures événementielles assurant le fonctionnement des interfaces.
Jeu d'essai	Les élèves imaginent un jeu d'essai vraisemblable afin de pouvoir tester le fonctionnement de l'application.
Tests et validation	Les élèves saisissent des données d'un jeu d'essai puis vérifient le bon fonctionnement des différentes interfaces.

Répartition des tâches

Les étapes de ce projet peuvent être réparties entre quatre élèves constituant le groupe.

La durée de travail de chaque élève est fixée à 16 heures.
Le diagramme de Gantt suivant montre une répartition possible des tâches.

tâches \ heures	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Rédaction :																
synthèse activité de l'organisation																
besoin d'évolution																
fonctionnalités attendues																
contraintes																
synthèse																
Cas d'utilisation																
saisie prescription																
consultation formations																
synthèse																
Evolution de la base de données																
Sécurisation de la base																
Initialisation des données de la base																
création des tuples de prescriptions suivies																
création des tuples de prescriptions non suivies																
Connexion à la base et tests																
Maquettage de l'application																
menu et prescription																
consultation formations																
Développement :																
page "saisie prescription"																
page "consultation"																
fonctions module																
intégration																
Tests :																
conception jeu d'essai																
tests																

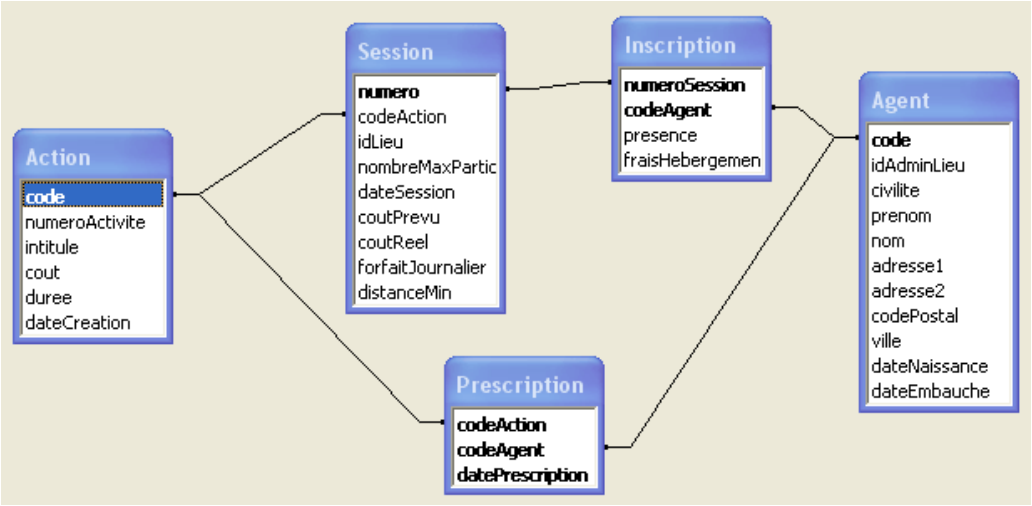


Éléments de solution

Modification de la base

Afin d'avoir un script de base de données avec des dates actualisées, les dates des sessions sont préalablement modifiées à l'aide de l'utilitaire proposé sur le site du Certa :
http://www.reseaucerta.org/outils/outils.php?num=371

Une nouvelle table « Prescription » est ensuite ajoutée dans la base de données :



La date de prescription mémorisée est la date du jour (cette donnée n'est pas visualisée dans la solution proposée). La durée de vie de la table « Prescription » étant de deux ans, sa clé permet ainsi au chef de service de renouveler une prescription qui n'aurait pas été suivie par l'agent concerné.

Un nouvel utilisateur avec des autorisations appropriées est ensuite ajouté à la base.

Création d'un jeu d'essai

On part de quelques critères statistiques proposés par les élèves, par exemple :
- 1/4 des inscriptions font suite à des prescriptions de chef de service ;
- 1/10 des prescriptions ne sont pas suivies par les agents.

Compte tenu du nombre d'inscriptions (environ 730) déjà présentes dans la base existante, il faut donc créer environ 180 prescriptions (730*1/4) correspondant à des inscriptions et 20 prescriptions (180*1/10) qui ne sont pas suivies d'inscription.

Le travail est réalisé sous PhpMyAdmin :

- création de la table

Exécuter une ou des requêtes sur la base bddFormation:

```
CREATE TABLE Prescription (
  codeAction CHAR(5) NOT NULL,
  codeAgent CHAR(8) NOT NULL,
  datePrescription DATE,
  PRIMARY KEY (codeAction, codeAgent, datePrescription),
  FOREIGN KEY (codeAction) REFERENCES Action(code),
  FOREIGN KEY (codeAgent) REFERENCES Agent(code));
```

- création aléatoire des 180 prescriptions (environ pour le jeu d'essai)

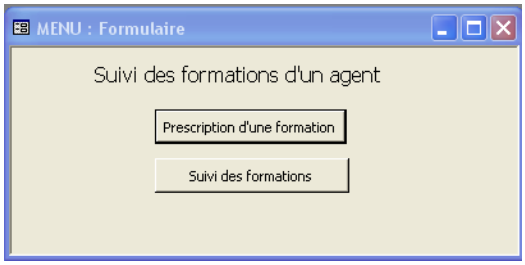
Exécuter une ou des requêtes sur la base bddFormation:

```
INSERT INTO Prescription
SELECT code, codeAgent, curdate()
FROM Action , Session , Inscription
WHERE code = codeAction
AND numero = numeroSession
and round(6*rand())<=1;
```

- création des 20 prescriptions sous forme de requêtes « insert into… »

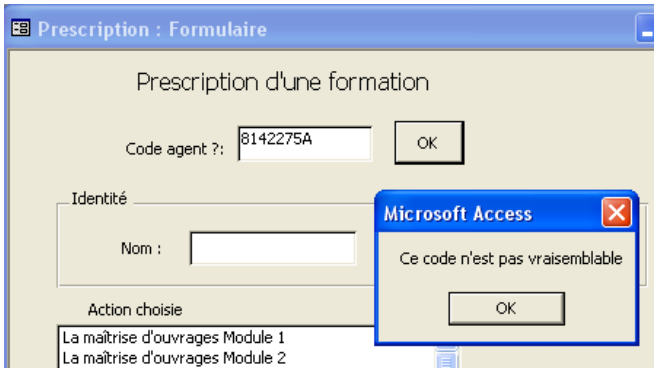
Maquettage

- menu principal

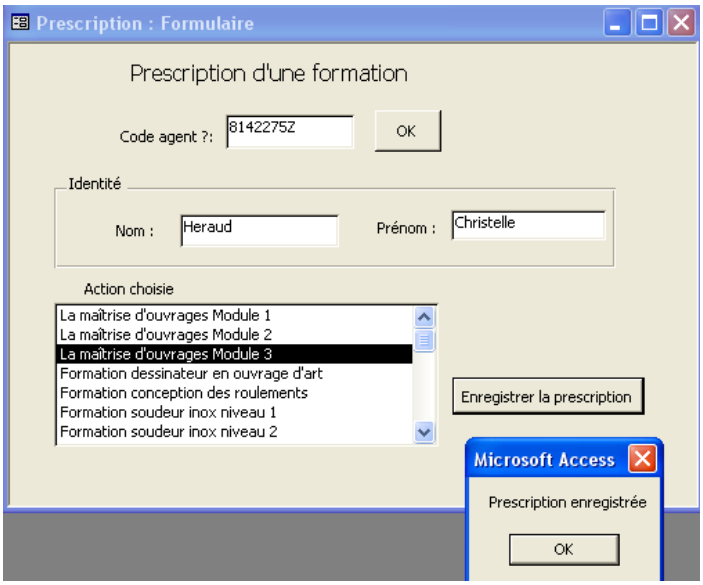


- saisie d'une prescription

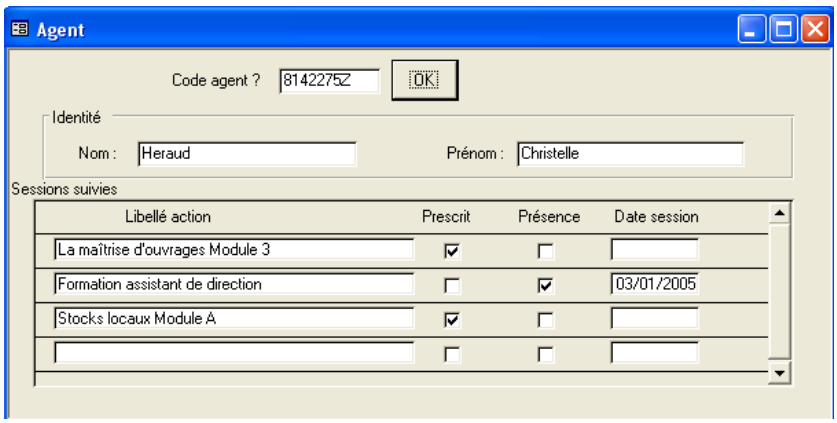
  - le code agent est erroné (contrôle local par modulo 23)



  - saisie correcte



- consultation des formations

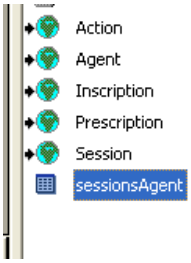


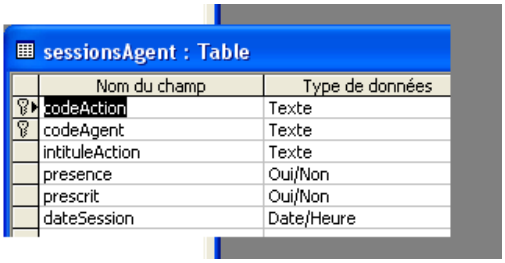
Connexion

On crée une connexion ODBC à la base de données MySQL en fournissant les données requises.

Structure des données locales

On crée une table temporaire locale pour traiter les inscriptions et prescriptions (nécessaire à la consultation des formations) :





Exemple de code pour l'interface d'affichage des formations (début)

```
Private Sub btnOK_Click()
  effaceTableLocale
  If codeVraisemblable(Me.codeAgentSaisi) Then
    Dim req As String
    req = "select nom, prenom from agent where code='" & Me.codeAgentSaisi & "';"
    Dim rs As DAO.Recordset
    Dim rsPres As DAO.Recordset
    '-- lecture données sur agent
    Set rs = CurrentDb.OpenRecordset(req)
    Me.nomAgentSaisi = rs("nom")
    Me.prenomAgentSaisi = rs("prenom")
    rs.Close
    req = "select code, codeAgent, presence, dateSession, intitule from action, session, inscription"
    req = req & " where code=codeAction and numero=numeroSession and codeAgent='" & Me.codeAgentSaisi & "';"
    '-- lecture des sessions suivies par l'agent
    Set rs = CurrentDb.OpenRecordset(req)
    DoCmd.SetWarnings False
    While Not rs.EOF    '-- pour chaque session suivie
      req = "select * from prescription where codeAgent='" & rs("codeAgent") & "' and codeAction='" & rs("code") & "';"
      '-- on vérifie si elle a été prescrite
      Set rsPres = CurrentDb.OpenRecordset(req)
      req = "insert into sessionsAgent values ('" & rs("code") & "','" & rs("codeAgent")
      req = req & "','" & doubleApostrophe(rs("intitule")) & "','" & rs("presence") & "',"
      If rsPres.EOF Then
        req = req & "0"
      Else
        req = req & "-1"
      End If
      req = req & ",'" & rs("dateSession") & "');"
      DoCmd.RunSQL req
      rs.MoveNext
    Wend
    rs.Close
    req = "select intitule, codeAction from prescription, action"
    req = req & " where code=codeAction and codeAgent='" & Me.codeAgentSaisi & "';"
```