

La programmation d'une application

La programmation de l'application de gestion des formations

Propriétés	Description
Intitulé long	Approche de la programmation avec le contexte de SI « Gestion des formations »
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	En jouant plusieurs rôles (utilisateur, informaticien), l'élève observe et manipule les composants d'une application : les données, l'IHM et le code associé.
Notions	2. L'adaptation de l'application 2.1. Prise en charge du dialogue homme-machine Interface homme-machine. - Maquette. - Ergonomie. - Contrôle graphique. - Événement 2.2. Exploitation des données Structure de données : tableau, indice, jeu d'enregistrements. 2.3. Programmation des traitements - Procédure, fonction, paramètre. - Structure de contrôle. - Mise au point de l'application.
Outils	Microsoft Access à partir de la version 2000, les outils du contexte 2, un navigateur. Le lecteur Flash Player. L'observation des différents écrans est proposée au travers d'une animation nécessitant, outre un navigateur, une extension Flash. Des boutons de navigation « suivant » et « précédent » simplifient la navigation et permettent de prendre tout le temps nécessaire à l'analyse de chacun des écrans des différentes animations.
Mots-clés	Bas de données, IHM, programmation
Pré-requis	Découverte du processus de gestion des formations
Lien(s)	
Auteur(es)	Jean-Philippe Pujol, Eric Deschaintre, Christian Draux
Version	1.0
Date de publication	Janvier 2007

Ce document présente une étude de l'application utilisée dans le contexte 2 : « Gestion des formations ».

Il s'appuie sur le cas d'utilisation « Créer une session » présenté dans la description générale du contexte.

Jouant le rôle de l'utilisateur « Responsable des formations », l'élève saisit les données nécessaires à la création d'une nouvelle session de formation.

Dans le rôle de l'informaticien, il est ensuite invité à observer les contenus des tables de la base nécessaires à ce traitement pour comprendre la provenance des données enregistrées lors de la création d'une session de formation.

Il est ensuite conduit à analyser le code VBA permettant, d'une part, de gérer le dialogue avec l'utilisateur, et, d'autre part, d'assurer l'enregistrement des données.

En prolongement de cette de séquence, l'élève est enfin invité à traiter quelques questions de compréhension en manipulant l'environnement de développement.

Le cas d'utilisation « Créer une session »

Utilisation de l'application

Rappel de la description textuelle du cas :

Cas d'utilisation	Créer une session
Acteur	Responsable des formations
Événement déclencheur	Début année ou besoin ponctuel
Parties prenantes et intérêts	Mettre à jour le planning des sessions afin de permettre l'inscription des agents
Portée	Planning des sessions
Pré-conditions	Composantes de la session valorisées et vérifiées ; responsable identifié
Scénario nominal	<ol style="list-style-type: none">1. Le système présente un formulaire2. Le responsable choisit l'activité concernée3. le système propose les actions correspondantes4. Le responsable choisit l'action concernée5. le système propose ses caractéristiques et les sessions concernées6. Le responsable renseigne le formulaire en saisissant la date, le nombre maximal de participants, le coût prévu et en choisissant le lieu. Puis il le soumet au système7. le système enregistre les données
Cas particuliers	7a. Certains champs sont mal renseignés : aller en 6.

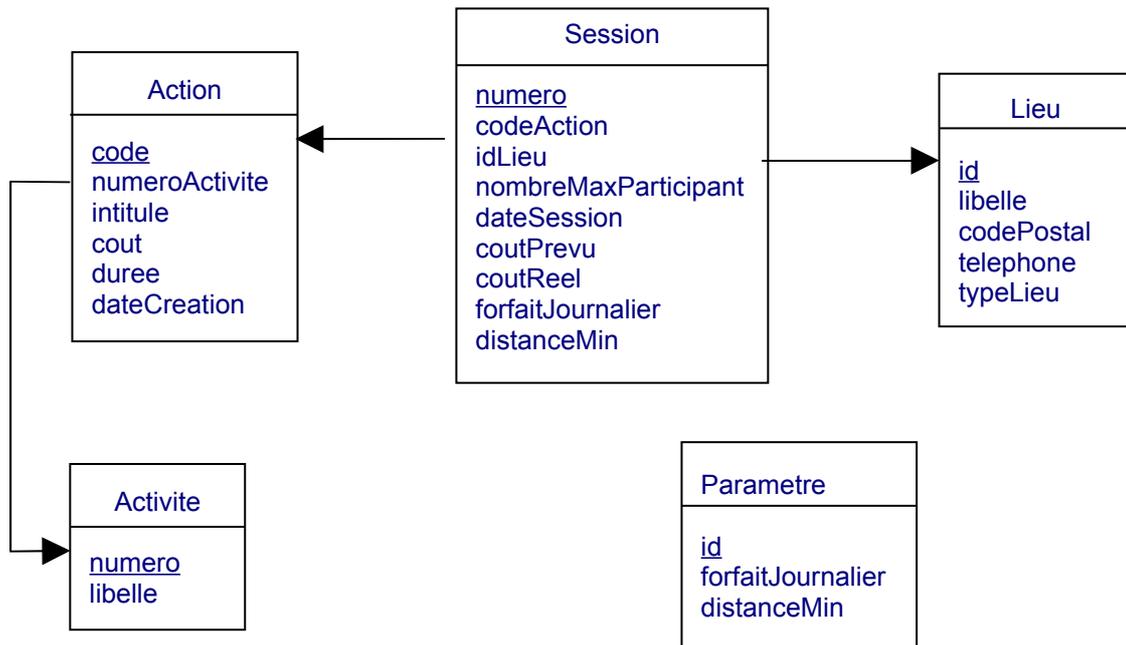
On se propose de saisir une nouvelle session dont les caractéristiques sont les suivantes :

- activité « gestion » ;
- action « Formation assistant comptable » ;
- lieu de la session : « Pau » ;
- nombre de personnes pouvant s'inscrire à la session : 8 ;
- date de la session : 15/01/2007 ;
- coût prévu : 365 €.

- Une démonstration *flash* permettant d'illustrer cette partie est associée à ce document : fichier <saisie.htm > (nécessite le fichier < saisie.swf>).

Observation des tables

La partie du schéma relationnel associée à ce cas d'utilisation est présentée ci-dessous.



On choisit l'activité à partir de son libellé et on utilise son numéro

numero	libelle
3	Matériel
4	Gestion
5	Voyageurs

On choisit l'action à partir de son libellé et on utilise sa clé qui est enregistrée dans le nouvel enregistrement de la table Session

code	numeroActivite	intitule	cout	duree	dateCreation
G0579	4	Contrôleur de gestion	1750	12	09/05/2006
G0580	4	Formation assistant comptable	7740	66	03/08/2006

numero	codeAction	idLieu	nombreMaxParticipant	dateSession	coutPrevu	coutReel	forfaitJournalier	distanceMin
167	E0581	12	8	03/01/2007	408		54	60
168	MD427	30	4	09/01/2007	361		54	60
169	G0580	18	8	15/01/2007	365		54	60

La clé de la nouvelle session est fournie par le système

Les données saisies sont enregistrées

Les valeurs actuelles des paramètres sont recopiées

Le lieu est choisi à partir de son libellé et sa clé est enregistrée

id	libelle	codePostal	telephone	typeLieu
17	St Jean Pied de	64450	0559003468	A
18	Pau	64000	0559333300	F
19	Dax	40100	0538642594	A
20	Léhon	40300	0538162595	A

id	forfaitJournalier	distanceMin
1	54	60

Un nouvel enregistrement est créé dans la table *Session* avec les données correspondant à la saisie. La clé est générée automatiquement par le système. Le *codeAction* G0580 provient de l'action sélectionnée dans la liste déroulante. L'*idLieu* provient de la table *Lieu* dans laquelle on a sélectionné Pau. Les données saisies sont enregistrées dans les champs correspondants. Les valeurs actuelles des paramètres, nécessaires ultérieurement pour les calculs des remboursements, sont recopiées depuis la table *Parametre*.

Analyse du fonctionnement

Contrôles et propriétés

Les données utilisées lors de l'enregistrement d'une nouvelle session proviennent soit d'autres tables, soit d'une saisie réalisée par l'utilisateur.

Comment se réalise l'insertion d'un nouvel enregistrement dans la table *Session* ? Comment les données sont-elles sélectionnées dans les différentes tables pour être transférées dans la table *Session* ? C'est le rôle du programme mis en œuvre lors de l'utilisation de l'interface. Ce programme est en fait constitué de différentes procédures qui réalisent les actions indiquées dans le cas d'utilisation sous la forme « le système présente... », « le système propose... », etc.

- Pour bien comprendre le fonctionnement de la partie programmation, il faut d'abord analyser l'organisation de l'interface de saisie, une démonstration *flash* permettant d'illustrer cette partie est associée à ce document : fichier <interface.htm > (nécessite le fichier < interface.swf>).

Synthèse :

L'interface contient différents contrôles (zone de texte, listes déroulantes, etc.) identifiés par un nom. Chaque contrôle possède différentes propriétés.

Un contrôle affichant des données peut être :

- un contrôle dépendant : son contenu provient directement de la base de données. Il reflète exactement le contenu d'un champ d'une table de la base. Cette table est celle dont dépend un formulaire ; ici, le sous-formulaire *session* a comme source de données une requête SQL figurant dans ses propriétés. Lorsqu'un contrôle est dépendant, il est le reflet du contenu d'un champ : écrire dans ce contrôle ou en modifier le contenu (si l'on dispose des autorisations nécessaires) revient à écrire dans la table sous-jacente ;
- un contrôle indépendant ; son contenu n'est pas lié à la base de données. Il provient d'une requête SQL, d'une saisie ou d'un programme (voir ci-dessous) ;
- un formulaire (ici le sous-formulaire *session*) dont le mode d'affichage continu permet d'afficher plusieurs sessions

Procédures événementielles

L'observation du comportement du formulaire lors de la saisie des données permet d'identifier les événements qui déclenchent un traitement :

- le fait de sélectionner une activité dans la liste déroulante *ldr_activite* (événement), déclenche la mise à jour de la liste déroulante des actions de formation (traitement);
- le fait de sélectionner une action dans la liste déroulante *ldr_action*, déclenche l'affichage des caractéristiques de l'action ainsi que les différentes sessions correspondantes;
- le fait de terminer la saisie des données d'une session (date, lieu, etc.), c'est à dire après avoir inséré un nouvel enregistrement dans la table *Session*, déclenche le calcul du numéro de la nouvelle session et son affichage;
- le fait de cliquer sur le bouton « Quitter », déclenche l'affichage du menu principal.

Numero	Date	Nb part.	Lieu	Coût héberg. prévu
14	20/02/2006	6	Bordeaux	619
134	25/09/2006	6	Bordeaux	892
169	15/01/2007	8	Pau	365

Les situations exposées ci-dessus sont des événements ayant un nom fixé par le logiciel Access :

Situation	Nom événement
Après avoir sélectionné...	AfterUpdate
En cliquant...	Click
Après avoir inséré un nouvel enregistrement...	AfterInsert

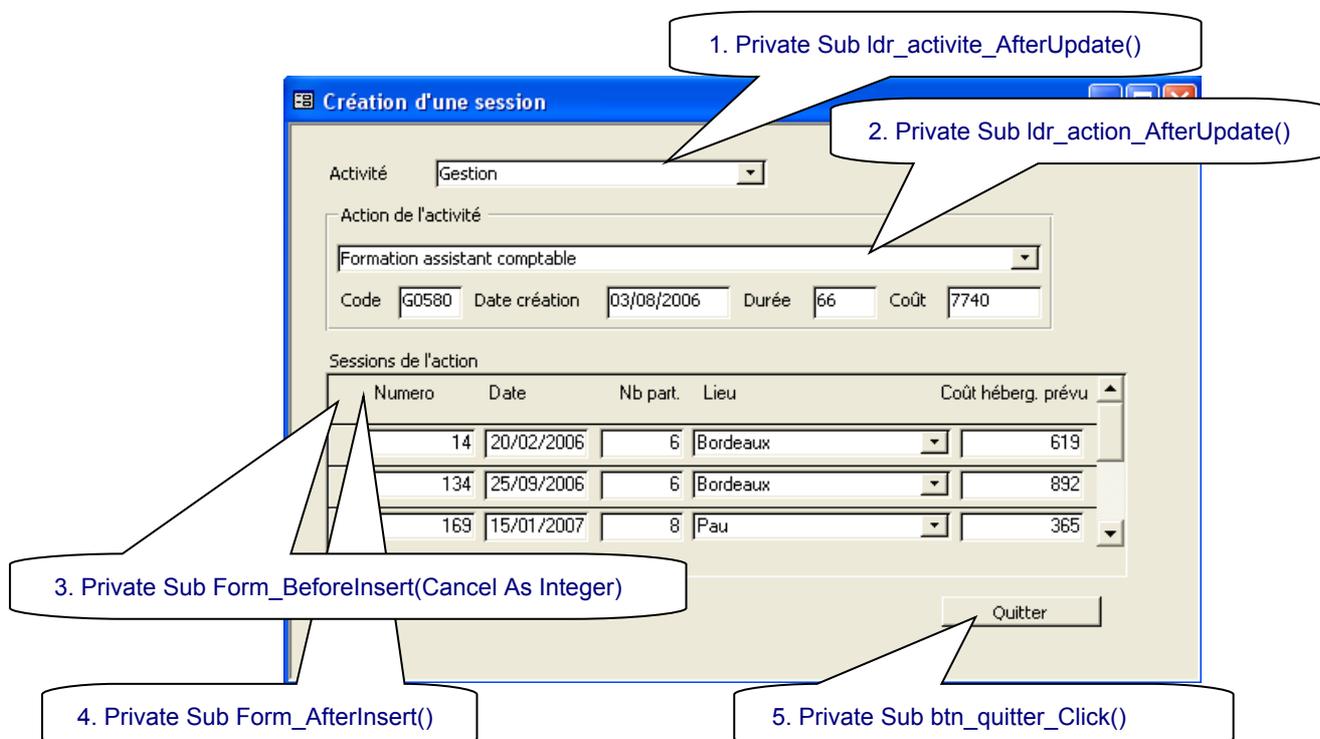
La gestion par programme d'un événement survenant sur un contrôle se fait par une procédure dont le nom, généré automatiquement, s'écrit sous la forme « nomDuContrôle_nomEvenement() ».

Ainsi, la procédure événementielle gérant l'événement « après avoir sélectionné une activité dans la liste déroulante « ldr_activite » se nomme ldr_activite_AfterUpdate()

- L'observation des différentes procédures événementielles permettant la gestion de cette interface se fait avec cette [animation <procedure.htm>](#) (nécessite le fichier <procedure.swf>.).

Synthèse :

Représentons l'interface en y indiquant les procédures événementielles supportées par les différents contrôles graphiques :



L'interface est composée de 2 parties :

- le formulaire principal « *gestionSession* », partie haute de l'interface qui permet de sélectionner les actions pour afficher l'action recherchée
- le sous-formulaire « *sessions* », représente la partie détails. Il permet la création et la modification des sessions. Les actions réalisées dans cette partie doivent respecter l'intégrité de la base de données

On peut interpréter par exemple la procédure « Private Sub ldr_activite_AfterUpdate() » ainsi :

- Private Sub : il s'agit d'une procédure
- Le nom de la procédure est : « ldr-activite_AfterUpdate »
- Le contrôle graphique qui déclenche l'évènement est la liste déroulante « ldr-activite »
- L'évènement qui déclenche le traitement (c'est-à-dire l'exécution de la procédure) est « AfterUpdate » (ce qui signifie : « après mise à jour de la valeur choisie par l'utilisateur »)

Les procédures sont numérotées pour indiquer l'ordre de mise en œuvre, conformément au cas d'utilisation.

Explications concernant le code utilisé :

1.

```
Private Sub ldr_activite_AfterUpdate()  
    Me.ldr_action.Requery  
End Sub
```

Lors de son exécution, cette procédure provoque une actualisation (« Requery ») du contrôle « ldr_action » appartenant à l'objet actuellement utilisé (le formulaire actif identifié par « Me »). Cette actualisation provoque l'exécution de la requête SQL présente dans la propriété « Contenu » de l'objet « ldr_action » : cette liste déroulante est remplie avec le résultat de cette requête.

2.

```
Private Sub ldr_action_AfterUpdate()  
    Dim rs As DAO.Recordset  
    Set rs=CurrentDb.OpenRecordset("select * from action where code='" &  
Me.ldr_action & "'")  
    Me.txt_codeAction = rs("code")  
    Me.txt_dateCreation = rs("dateCreation")  
    Me.txt_duree = rs("duree")  
    Me.txt_cout = rs("cout")  
    Me.sessions.Requery  
End Sub
```

Cette procédure fait exécuter une requête SQL retournant l'enregistrement de la table *Action* pour lequel le code est égal à la valeur choisie par l'utilisateur dans la liste déroulante *ldr_activite* (la valeur de la colonne 1 indiquée dans la propriété « Colonne liée »).

Le résultat de la requête est mémorisé dans un jeu d'enregistrement « rs » dont le contenu est ensuite affiché dans les contrôles graphiques

Ainsi par exemple, la zone de texte « txt_duree » du formulaire en cours d'utilisation (nommé « Me ») est remplie par le contenu du jeu d'enregistrement `rs("duree")`.

Après remplissage des différents contrôles, le sous formulaire « sessions » du formulaire actif « Me » est actualisé (« Requery ») afin d'afficher les sessions déjà enregistrées et de permettre la saisie d'une nouvelle session.

3.

```
Private Sub Form_BeforeInsert(Cancel As Integer)  
    Dim rsParam As DAO.Recordset  
    codeAction = Forms!gestionSessions!ldr_action  
    Set rsParam = CurrentDb.OpenRecordset("select * from parametre")  
    forfaitJournalier = rsParam("forfaitJournalier")  
    distanceMin = rsParam("distanceMin")  
End Sub
```

Avant de créer un nouvel enregistrement dans la table « session », on exécute une requête SQL retournant les valeurs des paramètres actuels contenus dans la table « parametre ».

Les valeurs présentes dans le jeu d'enregistrement sont copiées dans les contrôles visuels (« forfaitJournalier » et « distanceMin ») qui sont des contrôles dépendants : ces valeurs sont ainsi écrites dans le nouvel enregistrement de la table « session ».

4.

```
Private Sub Form_AfterInsert()  
    Me.Requery  
End Sub
```

Après avoir créé le nouvel enregistrement dans la table « session », le SGBD a généré une valeur pour la clé.

En actualisant le formulaire en cours, on provoque l'affichage de cette valeur dans le contrôle « numero » qui est un contrôle dépendant. L'utilisateur peut ainsi en prendre connaissance.

5.

```
Private Sub btn_quitter_Click()  
    DoCmd.Close  
End Sub
```

Cette procédure permet de fermer le formulaire actuellement en cours d'utilisation.

Travail à faire

1. Sur l'interface ci-dessous, indiquer les noms des différents contrôles graphiques utilisés.

Numero	Date	Nb part.	Lieu	Coût héberg. prévu
<input type="text"/>				

2. En observant les propriétés du contrôle « numero », expliquer pourquoi il n'est pas possible de le saisir ou de le modifier à l'aide du formulaire.

3. La liste déroulante `ldr_activite` est alimentée par une requête SQL qui retourne le numéro et le libellé des différentes activités. En observant les propriétés de cet objet, expliquer pourquoi ce numéro n'apparaît pas dans la liste déroulante.

4. Lors de l'ouverture du formulaire de saisie, on est invité à cliquer d'abord sur la liste déroulante « *Activité* ». Au lieu de commencer ainsi, que se passe-t-il si on clique en premier sur la liste déroulante « *Action de l'activité* » ? Expliquer pourquoi.

5. La personne en charge de la saisie des sessions de formation souhaite pouvoir choisir une activité également par son numéro (en plus du nom de l'activité). En effet il arrive qu'on ne lui communique que le numéro d'activité. Modifier l'interface de saisie d'une session pour afficher, à côté de la liste déroulante des activités, le numéro de l'activité sélectionnée (« *activite.numero* »)