

Introduction

Dans le cadre de l'unité de valeur d'initiation aux bases de données BD40, nous avons réalisé une base de données sous Access afin de gérer les événements organisés dans les locaux du Centre Culturel Franco Néerlandais du Château de Méridon, situé à Chevreuse dans les Yvelines (Ile de France).

I) Définition des besoins

La gestion du planning des événements est très délicate dans ce Centre Culturel car les événements sont très nombreux, de natures et de durées très diverses et se chevauchent parfois sur le plan temporel. S'assurer de l'absence de conflits dans l'utilisation des locaux tout en optimisant leur utilisation est un exercice délicat surtout lorsqu'il est effectué sans logiciel approprié.

La base de données que nous avons réalisé se propose donc de répondre aux attentes d'un tel Centre Culturel. Plus précisément, notre base de données est capable de gérer l'ensemble des événements, que ce soit des repas, des soirées ou des stages. Il est possible de connaître pour chaque période de la journée l'occupation des locaux, ainsi que de maintenir des listes d'inscrits et d'affectations dans les chambres dans le cas des stages. Une liste d'organismes permet de conserver une liste de contacts à relancer pour éventuellement organiser de nouveaux événements. Enfin, le logiciel peut gérer des listes d'actions déjà réalisées ou à réaliser afin de faciliter l'organisation des événements.

II) Conception

1) Le Modèle Conceptuel de Données (MCD)

La nature des événements organisés au Centre Culturel est très diversifiée. Afin de faciliter leur classement, nous avons dégagé trois types d'événements :

- les soirées,
- les repas,
- les stages.

Les soirées sont définies comme un événement occupant les locaux le soir et la nuit tandis les repas occuperont uniquement le midi. Le cas des stages est plus complexe car ils peuvent occuper divers moments de la journée en fonction des jours. Par exemple, l'organisateur d'un stage peut souhaiter réserver le Centre pour le déjeuner un jour N, tandis que le lendemain les inscrits seront en déplacement toute la journée et n'occuperont donc pas les locaux le midi. Cette très grande disparité dans les occupations des locaux doit être gérée finement pour permettre d'optimiser l'occupation des locaux. Ainsi, le Centre pourra accueillir un déjeuner d'un autre organisme pendant que les inscrits au stage sont en déplacement toute la journée.

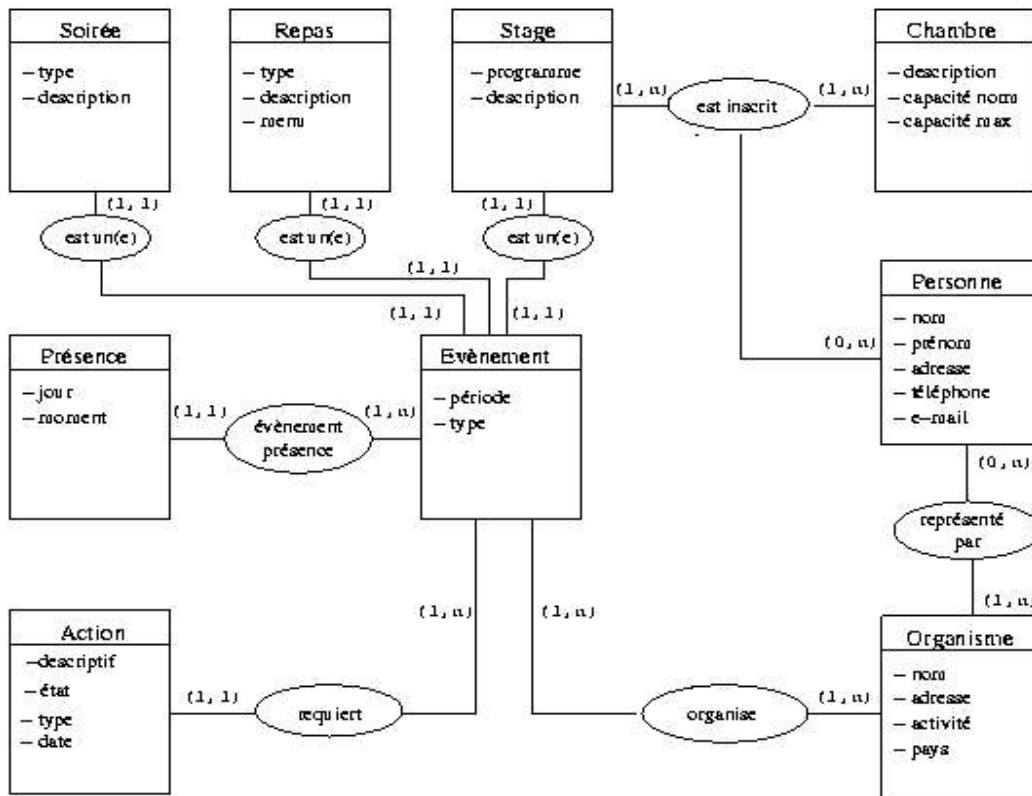
De ce constat, il est apparu naturel de découper chaque journée en quatre périodes :

- le matin,
- le midi,

- le soir,
- la nuit.

Chaque événement est donc soit un stage, soit une soirée, soit un repas, et il est lié à une liste de présences. Il est par ailleurs lié à une liste d'actions et un ou plusieurs organismes. Dans le cas où l'événement est un stage, il est lié à une liste des inscrits, chacun d'entre eux étant affecté ou non à une chambre. Pour chaque chambre, on connaît la capacité normale, et la capacité maximale (en rajoutant des lits supplémentaires). Chaque organisme est lié à une liste de personnes, représentant les contacts de l'organisme.

Modèle conceptuel de données

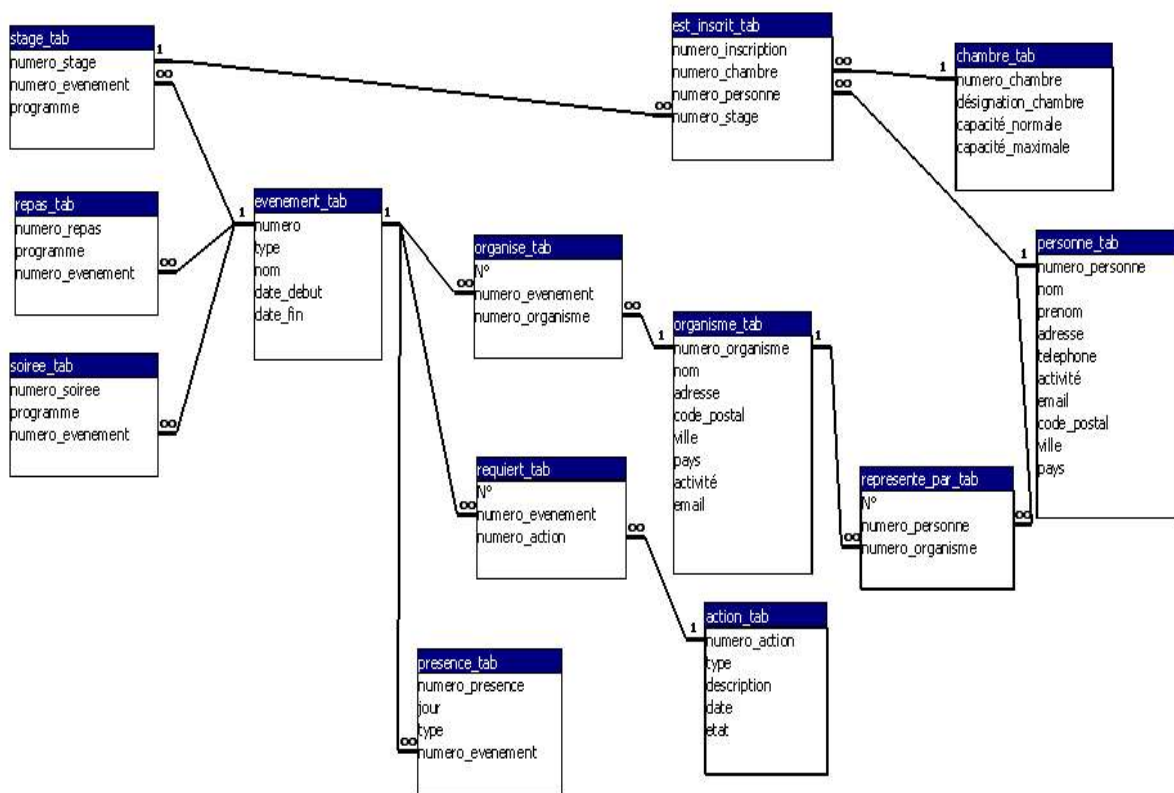


2) Le Modèle Logique de Données (MLD)

Les cardinalités entre les entités du MCD nous ont permis de supprimer plusieurs associations :

- les trois associations « est un(e) »
- l'association événement présence

Modèle Logique de Données



III) Réalisation

Dans le but de structurer au mieux la base de données nous avons réalisé la majorité des actions sur les formulaires en Visual Basic ou à l'aide de requêtes SQL.

1) Les formulaires

L'interface avec l'utilisateur se compose de 6 formulaires principaux :

- 1) le menu principal,
- 2) un formulaire pour gérer les stages
- 3) un formulaire pour gérer les repas,
- 4) un formulaire pour gérer les soirées,
- 5) un formulaire pour gérer les organismes,

- 6) un formulaire permettant de lancer diverses recherches sur les évènements et les actions.

Le *Menu principal* (*menu_form*) permet en un seul clic d'accéder aux autres formulaires.

- Le formulaire *Stage* (*stage_consulte_form*)

Ce formulaire a été conçu afin que l'utilisateur de la base puisse avoir tous les renseignements possibles sur l'ensemble des stages facilement et rapidement. De plus, il lui permet de modifier ou d'ajouter des stages de manière très simple. Pour cela, nous avons décidé de faire un formulaire comportant 5 onglets différents.

Nom	Date début	Date fin	Organisme
Séminaire aéronautique	07/01/03	10/01/03	Cockpit
Musik	12/07/03	18/07/03	Mme Blin

Onglet *Informations*

Dans cet onglet, l'utilisateur trouvera la liste de tous les stages et pour chaque stage : le nom du stage, la période à laquelle il aura lieu, le nom de l'organisme organisateur et le programme détaillé de celui-ci. A partir de cet onglet, on pourra ajouter un nouveau stage, modifier les informations concernant un stage ou bien encore supprimer un stage. Les deux premières actions sont réalisées à l'aide de deux formulaires indépendants (*stage_ajouter_form* et *stage_modifier_form*).

Dans un souci de cohérence de l'interface, les formulaires d'ajout et de modification d'un stage répondent au même design graphique que les formulaires d'ajout et de modification de soirées et de repas.

Microsoft Access - [stage_ajouter_form : Formulaire]
Fichier Edition Insertion Enregistrements Fenêtre ?

AJOUTER STAGE

Nom

Date début

Date fin

Organisme

Programme

Mode Formulaire NUM

Onglet Actions

Dans cet onglet, l'utilisateur pourra consulter la liste des actions liées au stage courant, ajouter une nouvelle action ou bien modifier une action sélectionnée. L'ajout ou la modification d'une action est réalisé grâce à deux formulaires indépendants (*action_ajouter_form* et *action_modifier_form*). Ces deux formulaires sont réutilisés pour l'ajout et la modification des actions à partir des formulaires Soirée, Repas et Recherche.

Microsoft Access - [action_modifier_form : Formulaire]
Fichier Edition Insertion Enregistrements Fenêtre ?

MODIFIER ACTION

Type

Date

Etat

Description

Mode Formulaire NUM

Onglet Chambres

Cet onglet permet à l'utilisateur de réaliser toutes les opérations concernant l'affectation des inscrits dans les chambres. Il dispose d'une liste des chambres avec le nombre de lits en temps normal, le nombre de lits maximal et le nombre de lits utilisés pour le stage courant. Il dispose pour chaque chambre de la liste des inscrits affectés à cette chambre et d'une liste des personnes inscrites au stage mais

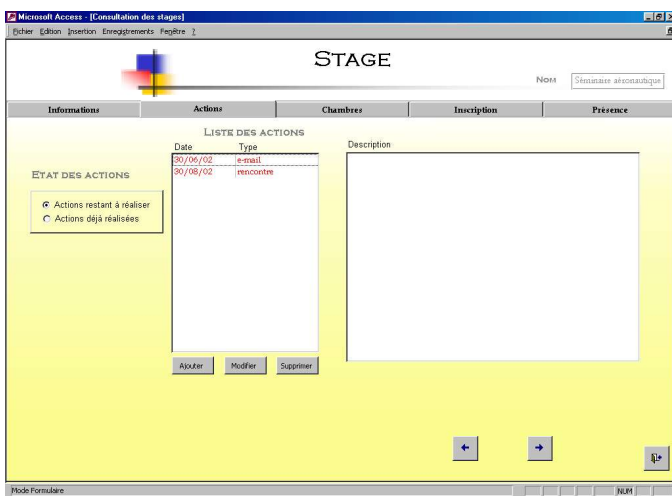
non affectées à une chambre. Il peut ajouter et retirer des personnes d'une chambre donnée.

Onglet *Inscription*

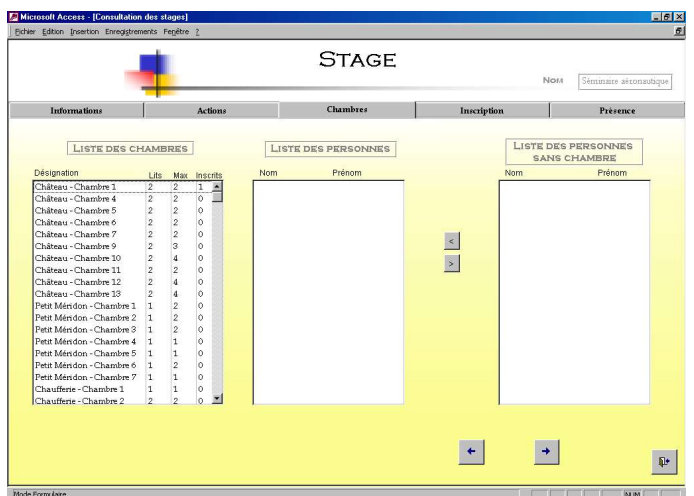
Cet onglet permet d'inscrire des personnes à un stage. L'utilisateur dispose de la liste des inscrits au stage, de la liste de toutes les personnes de la base et de la possibilité d'ajouter une nouvelle personne dans la base et de l'inscrire au stage courant.

Onglet *Présences*

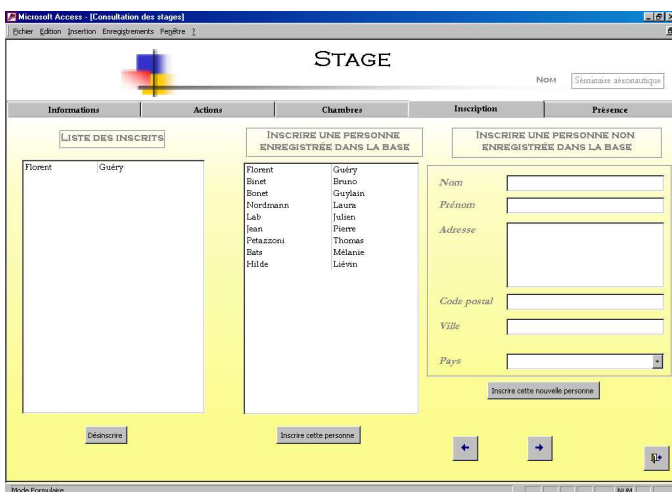
Cet onglet permet de consulter et de modifier les périodes de présence d'un stage. Lors de la création d'un stage, on considère que l'événement occupera les lieux durant toutes les périodes de la journée, tous les jours du stage. L'utilisateur peut ensuite modifier ces valeurs pour refléter l'occupation réelle des locaux.



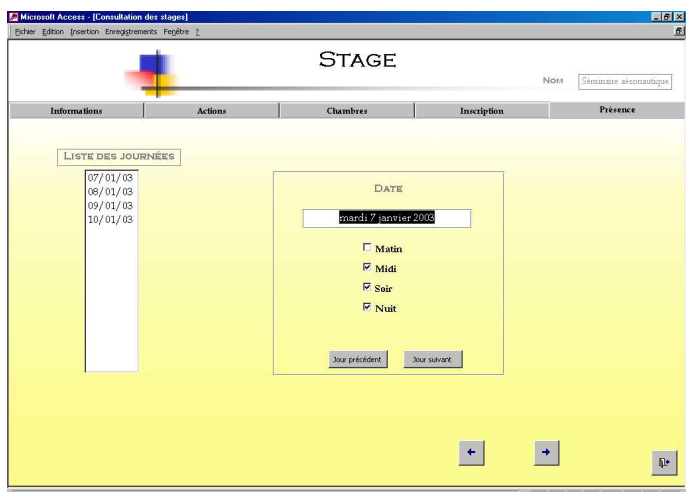
Onglet Actions



Onglet Chambres



Onglet Inscription



Onglet Présences

- Le formulaire *Repas*

Ce formulaire est similaire au formulaire Stage en ce qui concerne les informations et les actions. Il ne permet pas l'édition des présences, ni l'inscription des personnes ni l'affectation des chambres, ces trois possibilités n'ayant pas lieu d'être dans le cas d'un repas.

Nom	Date	Organisme
Alliance Française AG	08/01/08	Alliance Française

Onglet *Informations*

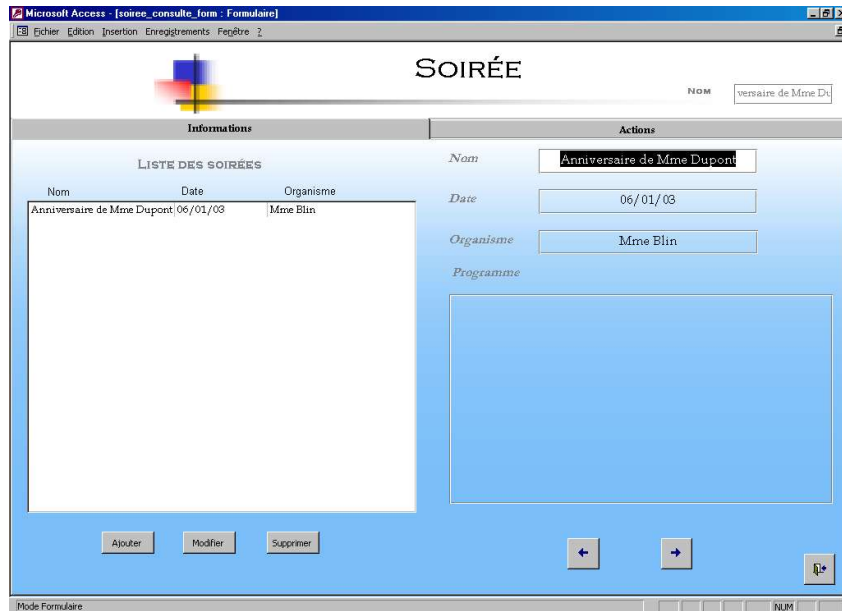
L'onglet information a été fait dans la même logique que l'onglet Informations pour les stages. Il contient le même type d'informations liées aux repas.

Onglet *Actions*

Cet onglet est de même type que celui présenté pour les stages.

- **Le formulaire *Soirées***

Ce formulaire permet à l'utilisateur d'obtenir des informations diverses sur les soirées et de gérer les actions pour chacune des soirées.



Onglet *Informations*

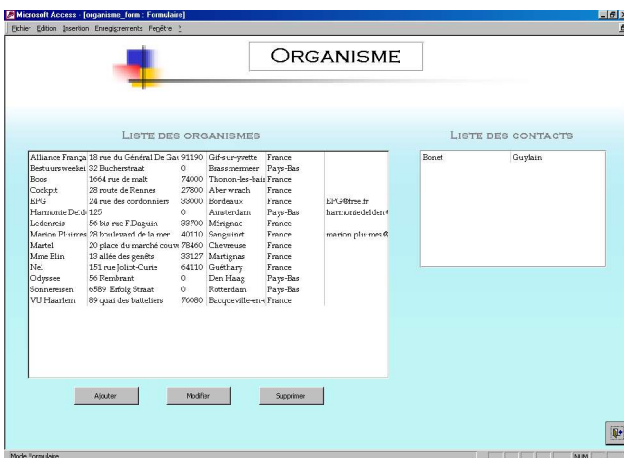
L'onglet information a été fait dans la même logique que l'onglet Informations pour les stages. Il contient le même type d'informations liées aux soirées.

Onglet *Actions*

Cet onglet est de même type que ceux présentés pour les stages et les repas.

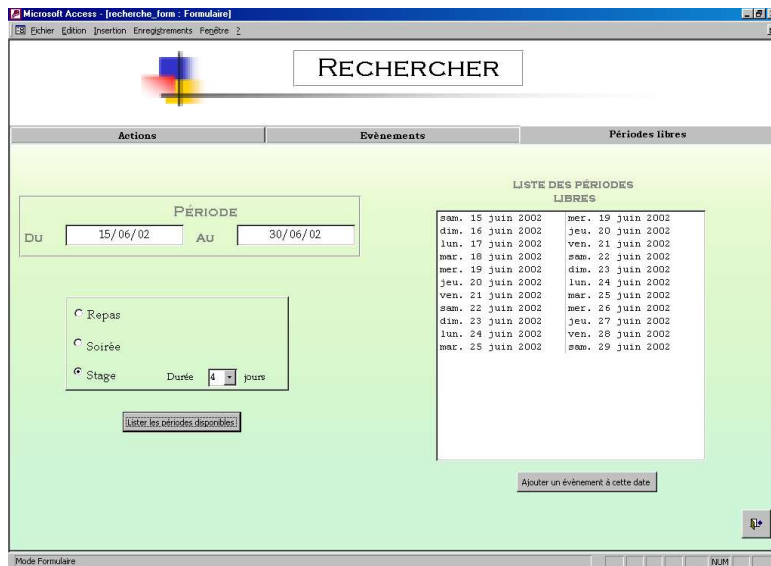
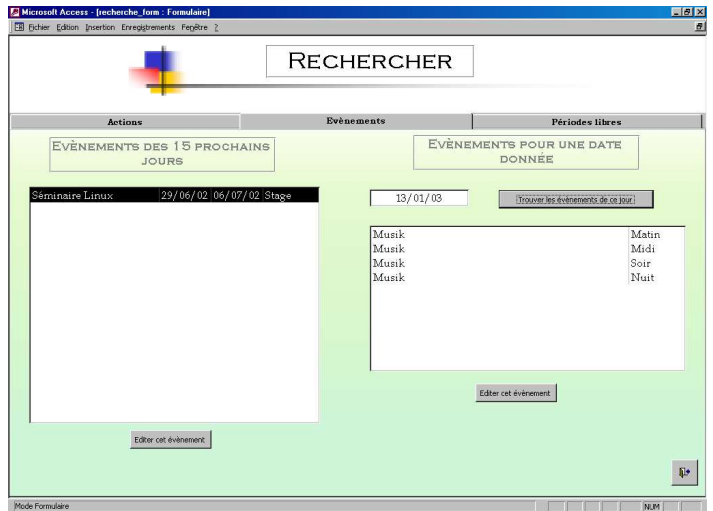
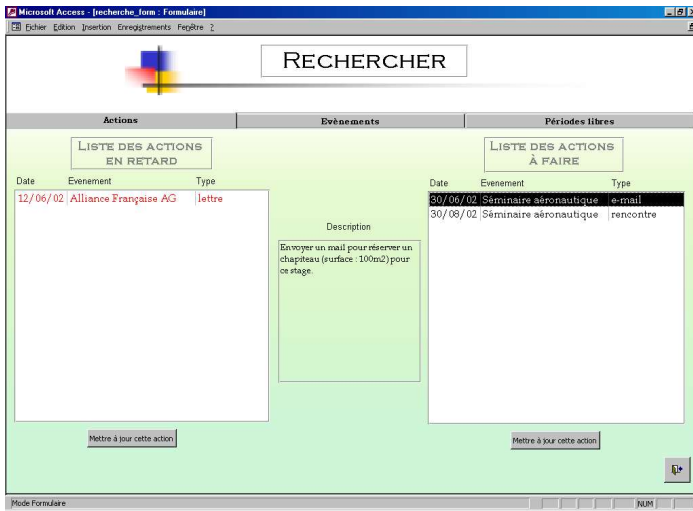
- Le formulaire *Organisme*

Ce formulaire donne une liste de tous les organismes enregistrés de la base. A partir de là, on peut grâce à deux formulaires indépendants ajouter et modifier des organismes. La modification d'un organisme permet de créer une liste de contacts pour l'organisme courant.



- Le formulaire *Recherches*

Ce formulaire se compose de trois onglets différents :



Le premier onglet donne deux listes d'actions : la liste des actions en retard (actions ayant du être réalisées avant la date courante) et la liste des actions à faire (liste des actions devant être réalisées à partir de la date courante). En sélectionnant une action, on peut accéder à un formulaire permettant de modifier toutes les informations concernant cette action.

Le deuxième onglet donne la liste des évènements ayant lieu dans les 15 prochains jours, et donne la possibilité de chercher les évènements organisés pour une date donnée.

Le troisième onglet permet de chercher des périodes pour organiser des évènements. On peut spécifier le type de l'évènement et la durée si l'évènement est un stage, ainsi que la plage temporelle de recherche. On obtient alors une liste de périodes libres que l'on peut sélectionner pour ajouter directement l'évènement.

2) Les requêtes SQL

Nous avons utilisé 14 requêtes en SQL au sein de la base afin d'accéder à des enregistrements particuliers.

- *lister_les_actions_a_faire_req* : Lister l'ensemble des actions restant à effectuer (à partir de la date courante).
- *chercher_inscrits_req* : Permet de lister toutes les personnes inscrites à un stage donné.
- *lister_actions_en_retard_req* : Lister l'ensemble des actions restant à effectuer, et qui sont en retard (avant la date courante).
- *lister_evenement_presence_15_prochain_jour_req* : Lister les événements des 15 prochains jours.
- *lister_evenement_present_pour_une_date_donnee_req* : trouve à partir d'une date donnée les événements présents ainsi que leurs périodes de présence.
- *organisme_lister_contact_req* : Lister tous les contacts d'un organisme donné.
- *organisme_lister_req* : Lister tous les organismes de la base.
- *repas_lister_req* : lister tous les repas.
- *soiree_lister_req* : lister toutes les soirées.
- *stage_chambre_req* : lister toutes les chambres ainsi que leur occupation pour le stage courant.
- *stage_lister_personnes_inscrites_req* : Lister toutes les personnes inscrites à un stage donné et affectées à une chambre donnée.
- *stage_lister_personnes_non_inscrites_req* : Lister les personnes inscrites à un stage donné mais non affectées à une chambre.
- *stage_lister_req* : lister tous les stages.
- *stage_nbr_personne_chambre_req* : lister le nombre de personne dans chaque chambre pour un stage donné.

Ces requêtes ne sont que des requêtes de sélection, qui ont été placés en dehors du code Visual Basic car elles sont utilisées à de nombreuses reprises. Les autres requêtes de sélection, d'insertion et de mise à jour (en nombre beaucoup plus grand) sont présentes dans le code Visual Basic, soit sous forme de requête SQL pure soit en utilisant directement l'interface DAO de Visual Basic.

3) Le code Visual Basic

La réalisation du logiciel a requis le développement de nombreuses fonctions en Visual Basic. Les fonctions sont présentes soit dans des modules indépendants, soit directement comme code événementiel dans les formulaires.

Pour programmer toutes ces fonctions, nous avons dû nous familiariser avec le langage Visual Basic, avec l'interface DAO pour accéder aux tables depuis Visual Basic ainsi qu'avec les possibilités de modification des propriétés des formulaires et des autres objets graphiques.

IV) Problèmes rencontrés

Notre premier problème majeur a été l'organisation graphique des formulaires: nous devons réaliser des actions complexes tout en proposant une interface graphique simple. Il a fallu donc réfléchir à ce problème avant de commencer la réalisation des formulaires et la programmation des évènements proprement dites

Il a fallu faire attention à ce que l'utilisateur ne puisse pas accéder à plusieurs formulaires simultanément. La solution vue en TP n'était pas satisfaisante car le formulaire de premier plan a parfois besoin d'accéder à des informations contenues dans le formulaire d'arrière-plan.

Nous n'avons pas pu égayer l'interface graphique comme nous l'aurions souhaité car dès que le nombre d'image par formulaire devient trop important, Access met beaucoup de temps à ouvrir les formulaires, et la base prend beaucoup de place disque.

V) Améliorations possibles

De nombreuses améliorations du logiciel sont possibles. Ainsi le système des présences n'a pas été utilisé de manière optimale. Il doit permettre de gérer les conflits entre les évènements, de les lister et de prendre en compte le fait que plusieurs évènements puissent avoir lieu simultanément (par exemple deux repas de 20 personnes peuvent avoir lieu le même midi). Par ailleurs, il est possible d'ajouter des fonctionnalités d'impression au logiciel (impression de la liste des inscrits, impression de la répartition dans les chambres, impression de la liste des évènements, des actions en retard, etc...). Le système d'inscription des personnes peut éventuellement être complété par un système permettant de gérer le règlement des stages.

Conclusion

En conclusion, ce projet nous a tout d'abord permis de découvrir un système de gestion de base de donnée : Access. Ensuite, nous avons pu découvrir comment concevoir notre base fin de gérer au mieux les informations contenues. Enfin, nous avons pu apprendre à effectuer des requêtes SQL complexes ainsi que la programmation en Visual Basic.