

TP N°1 : Création d'une base de données

Plusieurs questions vont apparaître au fur et à mesure du TP (précédées de *QUESTION* et encadrées), répondez-y individuellement sur feuille. Une correction orale vous sera donnée lors des prochaines séances de TP.

Si avez des questions au cours du TP (autres que celles dont on vous demande de répondre sur feuille), n'hésitez pas à les poser à votre chargé de TD.

Pensez à enregistrer régulièrement votre travail. Pensez aussi à sauvegarder votre travail sur une clé USB par exemple à la fin de chaque séance.

Rappel - les transparents de cours sont accessibles à l'adresse Web :

<http://www.lamsade.dauphine.fr/~ozturk/courses.html/>



Ou à Dauphine : P:\PROF\OZTURK\ETUD\OutilInformatique

L'objectif de ce TP est de vous faire créer une base de données, qui sera utilisée lors les TP suivants, pour exécuter des requêtes, créer des formulaires et des états, etc.

I. Prise en main d'Access

Dans cette première partie, vous allez vous familiariser avec Access en créant des relations (tables) dont on vous indiquera le schéma.

A. Lancer Access

- Access se lance à partir du menu *Démarrer, Tous les programmes, Office 2007*, en cliquant sur *Microsoft Access*.
- Si Access n'apparaît ni dans le menu, ni sur le bureau (icône ) , alors rechercher l'emplacement du fichier MSACCESS.EXE (Menu *Démarrer* et *Rechercher* – icône )
- Si malgré toutes les étapes précédentes, vous ne trouvez pas le logiciel, faites appel à votre chargé de TD.

B. Créer une base de données

1. Dans le fenêtre qui s'ouvre, choisir *Base de Données vide*
2. Le logiciel vous demande d'indiquer le nom de la nouvelle base de données. Votre nouvelle base de données devra s'appeler `BD_Test_TD1.mdb` et **devra être sauvegardée dans votre répertoire personnel** (si vous le stockez sur le disque local il sera supprimé).

① Les fichiers de base de données sous Access ont pour extension : `.mdb`

C. Créer une relation (table)

1. Une fenêtre s'affiche. Par défaut, une table nommée *Table1* est ouverte.

Choisir le mode création (en cliquant sur l'équerre et le compas en haut à gauche, ou en choisissant *Affichage – Mode création*). Nommez cette table *Etudiants*.

2. Une fenêtre s'affiche vous permettant de saisir le schéma de votre nouvelle relation *Etudiants* – voir Figure 1.

La partie quadrillée supérieure vous permet de saisir le nom des différents attributs (ou champs), de sélectionner leur type de données dans une liste, et le cas échéant de donner une description de l'attribut.

La partie en dessous va se mettre à jour en fonction des différents types de données que vous sélectionnez. Faites attention à la partie droite de la partie inférieure. Des messages d'aide y sont régulièrement affichés.

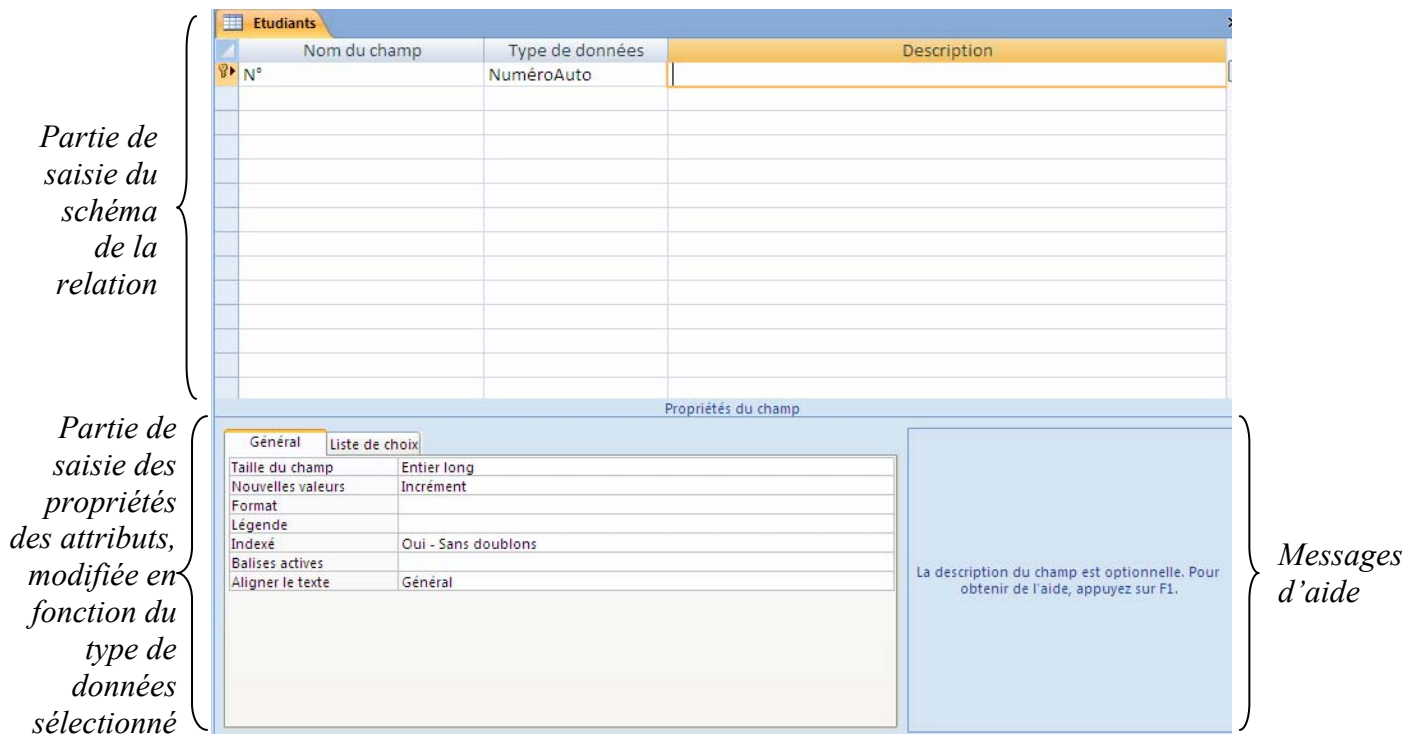


Figure 1 - Fenêtre affichée lors de la création de votre première relation nommée par défaut *Table1*.

- a) Créer un premier attribut intitulé *NumeroCarte*.
 - ① Prenez l'habitude de créer de noms d'attributs simples et compréhensibles. Evitez d'y inclure des espaces et des accents (la majorité des SGBD ne les acceptent pas autant prendre l'habitude de ne pas en mettre tout de suite – même si *Access* l'autorise). Vous pourrez modifier le nom de l'attribut, pour l'affichage uniquement, en mettant à jour la propriété *Légende*.
- b) Sélectionner le type *Numérique* pour cet attribut – La partie inférieure se met automatiquement à jour. Cet attribut doit être un entier long, sans partie décimale (sélectionner 0 dans la propriété *Décimales*).
- c) Inscrire *Numéro de carte d'étudiant* dans la propriété *Légende*. Le texte *Numéro de carte d'étudiant* sera ainsi affiché à l'utilisateur à la place du nom interne de l'attribut *NumeroCarte*.
- d) L'attribut *NumeroCarte* doit être la clé primaire de la relation. Ceci est fait par défaut par *Access* : vous pouvez voir le dessin d'une clé à gauche du nom du champ. Si nous voulions changer ceci, il suffirait de cliquer sur l'icône *clé primaire* dans la partie haute de la fenêtre.

Observer que lorsque l'on change le statut de champ (clé primaire ou non), la propriété *Indexé* est mise à jour : *Oui-Sans doublon*¹ lorsque le champ est clé primaire, *Non* lorsque le champ ne l'est pas.

QUESTION 1 : Pourquoi la propriété *Indexé* est-elle mise à jour lors de la définition de la clé primaire ?

① En principe, vous devriez préciser que vous n'acceptez pas les valeurs *null* pour cet attribut mais par défaut une clé primaire ne peut pas avoir de valeur *null*, donc il est inutile de le préciser (c'est un petit bug d'affichage d'Access, mais cette contrainte sera bien vérifiée lors de l'insertion des nuplets – on le verra plus tard).

- e) Créer un deuxième attribut intitulé *Nom* de type *Texte* et réduire la taille du champ à 30 (inutile de conserver 255 caractères pour un nom de famille).
- f) Cliquer dans la propriété *Format* et appuyer sur la touche F1 de votre clavier pour afficher l'aide. Après lecture de l'aide (en choisissant la première suggestion), saisir le format de telle sorte que le nom de famille soit forcément affiché en majuscules.

QUESTION 2 : Quel format de texte permet de forcer les caractères majuscules (quelle que soit la saisie de l'utilisateur) ?

L'attribut *Nom* va contenir le nom de famille des étudiants.

QUESTION 3 : Peut-on autoriser (1) que l'attribut prenne la valeur *null*² et (2) que l'attribut prenne la valeur *chaîne vide*³ ? Expliquer.

En fonction de vos réponses à la question précédente, mettre à jour les propriétés *Null interdit* et *Chaîne vide autorisée*.

- g) Dans la propriété *Indexé*, sélectionner *Oui - Avec doublons*.

QUESTION 4 : Quel est l'intérêt que cet attribut soit indexé et pourquoi *Avec doublons* ? Expliquer.

- h) Fermer la fenêtre de création de la relation (table) en cliquant sur la croix (✖).
- i) Dans la partie gauche de la fenêtre de base de données double-cliquer sur la relation *Etudiants*.

La relation (table) est ici ouverte en mode saisie/visualisation des nuplets. Il s'agit lors de cette étape de tester les deux attributs précédemment créés pour vérifier que tout fonctionne correctement.

① Pensez à réaliser des tests régulièrement en saisissant un ou deux nuplets que vous supprimerez par la suite, afin de vérifier que vous avez correctement saisi les propriétés. Ne saisissez jamais tous les attributs d'un coup pour les tester à la fin, vous pourriez perdre beaucoup de temps pour trouver vos erreurs. En informatique, **il faut toujours y aller par étapes et tester régulièrement !!**

¹ Il y a des doublons pour un attribut lorsqu'une même valeur apparaît plusieurs fois pour cet attribut dans différents nuplets.

² La valeur *null* est une valeur particulière en bases de données indiquant *sans valeur* ou *valeur inconnue*.

³ Une chaîne de caractère vide correspond à la valeur "" (sans espace – espace étant un caractère), i.e. la valeur chaîne sans caractère. Il s'agit d'une valeur. La valeur chaîne vide est différente de la valeur *null*.

Insérer plusieurs nuplets tests :

- Insérer un nuplet ayant uniquement une valeur de numéro de carte d'étudiant et quitter la fenêtre.

QUESTION 5 : Que se passe-t-il ? Expliquer pourquoi.

- Insérer la valeur du nom en vérifiant qu'elle s'affiche bien en majuscule.
- Insérer un autre nuplet avec la même valeur de carte d'étudiant que celle saisie précédemment, et saisir une valeur pour l'attribut *Nom*.

QUESTION 6 : Que se passe-t-il ? Expliquer pourquoi.

- Comme nous allons continuer à saisir le schéma de la relation, penser à supprimer les nuplets tests (les sélectionner tous et faire supprimer dans la barre du haut). Quitter la fenêtre.

① Il est préférable de supprimer les nuplets précédemment insérés. La modification du schéma de la relation (la définition des attributs de la table) est toujours plus facile quand il n'y a pas de nuplet, d'autant plus lorsque des attributs n'acceptant pas la valeur *null* sont ajoutés.

Pour supprimer un nuplet (enregistrement) sélectionner-le (en cliquant au début de l'enregistrement – partie grisée) et avec le bouton droit sélectionner le menu *Supprimer l'enregistrement*. Pour en supprimer plusieurs en même temps, appuyer sur la touche *ctrl* pendant la sélection. Pour supprimer tous les nuplets (enregistrements) d'une relation (table), cliquer en haut de la relation (triangle grisé sur la ligne où apparaissent les noms d'attributs – tous les nuplets vont être sélectionnés) et avec le bouton droit de la souris, sélectionner le menu *Supprimer l'enregistrement* (vous aurez un message vous indiquant que plusieurs enregistrements vont être supprimés).

Attention : Effacer les valeurs des attributs ne supprime pas l'enregistrement. Cela affecte uniquement des valeurs *null* aux attributs dont vous avez effacé les valeurs pour l'enregistrements

- j) Revenir en mode *Création* en cliquant sur la relation *Etudiants* avec le bouton droit de la souris, menu *Mode Création* (ou cliquer sur l'icône d'affichage dans la barre du haut).
- k) Créer un troisième attribut intitulé *Prenom*, de type *Texte* et de taille 20, de *Légende Prénom* (avec un accent).
- l) Inscrire dans la propriété *Masque de saisie*, la chaîne suivante `>L<L????????????????`, composée du symbole supérieur pour imposer que la première lettre soit en majuscules, d'un L majuscule (pour forcer la saisie d'un caractère de A à Z), du symbole inférieur, pour imposer que les caractères suivants soient en minuscules et de 19 points d'interrogation pour indiquer que les caractères suivants sont facultatifs.
Vérifier dans l'aide et en particulier dans les exemples fournis dans l'aide que ce format est correct.
- m) Mettre à jour les propriétés *Null interdit* et *Chaîne vide autorisée* de l'attribut *Prenom*.
- n) Insérer quelques nuplets pour tester que les propriétés de l'attribut *Prenom* sont bien vérifiées.

- o) Revenir en *Mode Création* et ajouter un attribut intitulé *Sexe*, de type *Texte* et de taille 1.
- p) Cliquer sur l'onglet *Liste de choix* pour cet attribut et sélectionner *Zone de liste* dans la propriété *Afficher le contrôle*. Sélectionner *Liste valeurs* pour la propriété *Origine source*. Dans la propriété *Contenu*, saisir la chaîne "F"; "M" qui représente le contenu de la liste de valeurs. Vous venez de créer une liste de deux valeurs *F* et *M*.
Interdire la valeur *null* et les chaînes vides.
Fermer la fenêtre et insérer un ou deux nuplets pour tester ce nouvel attribut.
Si les nuplets précédemment créés n'ont pas été supprimés, un message s'affiche indiquant un problème pour les nuplets de la table (le champ *Sexe* n'est évidemment pas renseigné). Choisir de les tester avec la nouvelle règle, puis de garder la vérification avec la nouvelle règle. Renseigner ensuite le champ *Sexe* pour les nuplets existants.
- r) Revenir en *Mode Création* et ajouter un attribut *DateDeNaissance*, de type *Date/Heure*, de format *date, abrégé*, avec comme valeur *Date de naissance* pour la propriété *Légende*. Cet attribut ne doit pas accepter les valeurs *null*.
- s) On souhaite que toutes les dates soient saisies sous la forme *jj/mm/aaaa*, c'est-à-dire deux chiffres pour le jour, deux chiffres pour le mois et quatre chiffres pour l'année, ex. *19/09/2006*.

QUESTION 7 : (1) Quelle propriété faut-il mettre à jour pour imposer la contrainte précisée au-dessus et (2) quelle est la valeur de cette propriété ?

QUESTION 8 : Pourquoi crée-t-on un attribut <i>DateNaissance</i> et non pas un attribut <i>Age</i> ? Expliquer.
--

- t) Insérer quelques nuplets pour tester ce nouvel attribut.
- u) Revenir en *Mode Création* et ajouter un attribut *Adresse* de type *Texte*.
Dans la propriété *Format*, saisir : @;"Inconnue". Ce format va permettre d'inscrire automatiquement *Inconnue*, si aucune valeur n'est saisie ou si la chaîne saisie est vide.
Insérer quelques nuplets pour tester ce nouvel attribut.

D. Lier plusieurs relations

Les étapes de cette section vont vous permettre de créer une nouvelle relation *Telephones* contenant un attribut *RefEtudiant*, un attribut *Numero* et un attribut *TypeTelephone*. Cette relation va contenir les numéros de téléphone des étudiants (ex. Fixe et Portable).

Il aurait été possible d'ajouter deux nouveaux attributs dans la relations *Etudiants* (ex. *NumeroTelFixe* et *NumeroPortable*), mais cette solution a plusieurs inconvénients : (1) Si un étudiant n'a pas de téléphone fixe ou portable, il y aura un attribut de valeur *null* (donc perte d'espace⁴). (2) Si de nouveaux types de téléphones sont créés (ex. *TelephoneDesParents*), il faudra ajouter de nouveaux attributs à la relation.

Il a donc été choisi de créer une nouvelle relation qui va être associée à la relation *Etudiants*. Cette solution a l'avantage d'être plus souple que la précédente car il est possible d'ajouter autant de types de téléphone que souhaité. Elle a néanmoins un inconvénient : lorsqu'un utilisateur

⁴ Car quelle que soit sa valeur, l'attribut occupe la taille maximum indiquée dans la propriété *Taille du champ*.

souhaitera voir toutes les informations concernant un étudiant, il devra faire une requête regroupant les données des deux relations *Etudiants* et *Telephones* (cf. prochaine feuille de TP).

❶ **Avant de créer le modèle relationnel de votre base de données, il faut donc bien réfléchir et peser le pour et le contre des différents choix possibles en fonction du problème posé, des besoins des utilisateurs, des contraintes etc.**

Dans cette section, le choix s'est porté sur une nouvelle relation (table) pour vous montrer comment lier deux relations.

1. Créer une nouvelle relation (table), à l'aide du menu *Créer* dans la barre supérieure.
2. Créer un attribut *RefEtudiant* de même type, même taille et de mêmes propriétés *Décimales* et *Valeur par défaut* que l'attribut *NumeroCarte* de la relation *Etudiants* créée précédemment. Faire en sorte que l'attribut soit indexé avec doublons.

QUESTION 9 : Pourquoi choisit-on d'indexer avec doublons et non pas sans doublon ?

3. Saisir dans la *Description* de cet attribut les informations suivantes : « *Correspondant au numéro de la carte d'étudiant - Clé étrangère faisant référence à l'attribut NumeroCarte de la relation Etudiants* ». Cette phrase est un commentaire pour vous et pour toute personne amenée à manipuler votre base de données ultérieurement.

❶ L'attribut *RefEtudiant* n'est pas encore défini en tant que clé étrangère, il va l'être par la suite (voir item 10).

4. Enregistrer la table en faisant un clic droit sur le nom de la table (au dessus des champs).

❶ Un message vous indiquant que vous n'avez pas créé de clé primaire apparaît. Le logiciel vous propose d'en créer une pour vous (**une relation doit toujours avoir une clé primaire !**), qui sera un numéro incrémenté automatiquement (type *NumeroAuto* et nouvelles valeurs *Incrément*).

Si vous avez malencontreusement cliqué sur *non*, vous pouvez faire à la main ce qui aurait été fait automatiquement si vous aviez cliqué sur *oui*. Créer un attribut, nommé par exemple *TelephoneID*, de Type *NumeroAuto*, et définissez cet attribut comme clé primaire.


5. Créer un attribut *Numero* de type *Texte*, de taille 14, n'acceptant pas la valeur *null* ni les chaînes vides. Créer un masque de saisie imposant que le numéro de téléphone soit de la forme : 5 couples de chiffres séparés par un espace (ex. 01 44 05 44 05). Si la question est posée, choisir d'enregistrer sans les symboles dans le masque.

QUESTION 10 : Quel est le masque de saisie ?

6. Tester en insérant quelques nuplets.

Au lieu de choisir la clé primaire automatique proposé par Access, il aurait également été possible de choisir une clé primaire multi-attributs, composée du couple d'attributs⁵ (*RefEtudiant,Numero*).

QUESTION 11 : Si on avait choisi de prendre pour clé primaire le couple d'attributs (*RefEtudiant,Numero*), aurait-il été possible d'associer plusieurs numéros de téléphones à un même étudiant ? Expliquer.

⁵ Il est possible de créer une clé primaire multi-attributs en sélectionnant les attributs appartenant à la clé et en cliquant sur l'icône  de la barre d'outils en haut.

QUESTION 12 : Même question que précédemment, si on avait choisi de prendre comme clé primaire uniquement l'attribut *RefEtudiant* (clé mono-attribut) ? Expliquer.

QUESTION 13 : Peut-on associer un même numéro de téléphone à différents étudiants si on choisit de prendre comme clé primaire uniquement l'attribut *Numero* ? Expliquer.

7. Revenir en *Mode Création* et ajouter un attribut *TypeTelephone*, de type *Texte*, de taille 10, n'acceptant pas la valeur *null*, ni les chaînes vides.
8. Cliquer dans l'onglet *Liste de choix* pour cet attribut et indiquer qu'il s'agit d'une *Zone de liste* (pour permettre l'ajout de types de téléphone), d'*Origine source* liste de valeurs, et de *Contenu* : "*Fixe*"; "*Portable*".
9. Tester en insérant quelques nuplets.
10. Les relations vont maintenant être liées (associées). Pour cela, il faut d'abord fermer les deux tables. Ensuite, cliquer sur le menu *Outils de base de données*, puis sur l'icône *Relations*. Une fenêtre s'affiche vous demandant de choisir des tables. Ajouter les deux relations (tables) *Etudiants* et *Telephones*, créées précédemment. Elles apparaissent dans la fenêtre intitulée *Relations*⁶ (vous pouvez fermer la fenêtre *Ajouter tables*). Cliquer sur l'attribut *RefEtudiant* de la relation *Telephones* en maintenant votre doigt sur le bouton gauche de la souris et glisser la souris (en maintenant votre doigt sur le bouton gauche) vers l'attribut *NumeroCarte* de la relation *Etudiants*. La fenêtre suivante doit s'afficher :

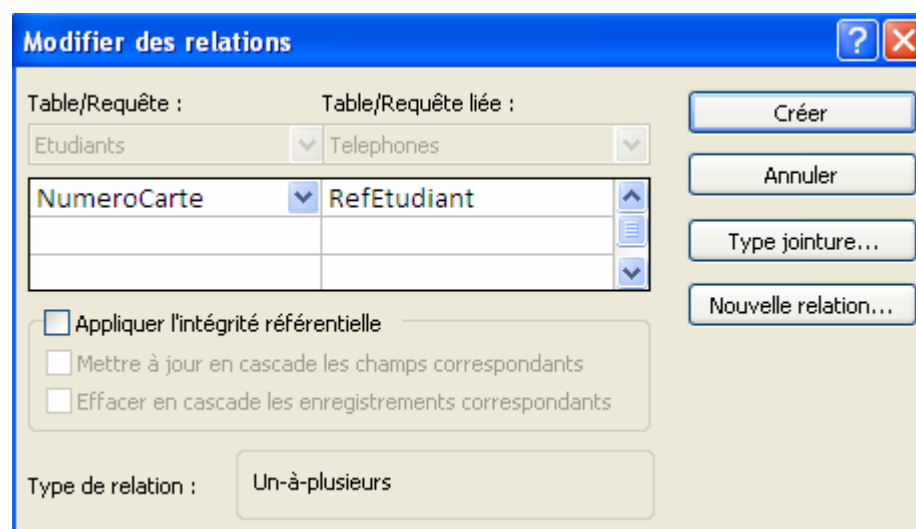


Figure 2 - Fenêtre de définition de clé étrangère.

11. Cocher *Appliquer l'intégrité référentielle* et cliquer sur le bouton *Créer*.
 - ① Si vous n'avez pas supprimé les nuplets tests des étapes précédentes, vous risquez d'avoir un message d'erreur. Supprimez-les et recommencez les items 10 et 11).
12. Insérer un nuplet test dans la relation *Telephones* dont la valeur *RefEtudiant* ne correspond à aucune valeur de l'attribut *NumeroCarte* de la relation *Etudiant*.

⁶ **Rappel** : Access utilise un vocabulaire incorrect, le terme *Relations* d'Access correspond en réalité à *Associations*. Une relation (du modèle relationnel) est une table.

Si vous avez bien coché *Appliquer l'intégrité référentielle*, vous devez obtenir un message d'erreur (sinon retourner dans la fenêtre Relations, cliquer sur l'association entre les deux tables et reprenez à l'item 11).

QUESTION 14 : Pourquoi avez-vous ce message d'erreur ?

① Dans un prochain TP, vous verrez comment faire en sorte que, lors de la saisie de la valeur de l'attribut *RefEtudiant*, les valeurs de l'attribut *NumeroCarte* de la relation *Etudiants* y apparaissent sous forme de liste.

13. Créer un nuplet de la relation *Etudiants* et ajouter deux nuplets associés dans la relation *Telephones*.

① Si vous retournez dans la relation *Etudiants* (en mode ajout de nuplet), vous verrez qu'un + apparaît devant le nuplet correspondant. Si vous cliquez sur ce symbole, les nuplets correspondants de la relation *Telephones* seront affichés. Il est même possible (grâce à l'association entre les relations) d'ajouter de nouveaux numéros de téléphones à un étudiant, en passant par la relation *Etudiants*.

14. Essayer maintenant de supprimer le nuplet ajouté précédemment dans la relation *Etudiants*. Vous devez obtenir un message d'erreur.

QUESTION 15 : Pourquoi obtenez-vous un tel message d'erreur ? Expliquer.

QUESTION 16 : Comment pouvez-vous faire pour que la suppression d'un nuplet dans *Etudiants* entraîne la suppression des nuplets correspondants dans la relation *Telephones* ?

15. Essayer maintenant de modifier la valeur de l'attribut *NumeroCarte* du nuplet ajouté précédemment dans la relation *Etudiants*. Vous devez obtenir un message d'erreur.

QUESTION 17 : Comment pouvez-vous faire pour que la mise à jour de l'attribut *NumeroCarte* d'un nuplet dans *Etudiants* entraîne la mise à jour des nuplets correspondants dans la relation *Telephones* (ex. pour modifier chaque année le nouveau numéro de carte d'étudiant) ?

II. A vous de jouer !

Vous connaissez maintenant les principales fonctionnalités permettant de créer une base de données. Cette section va vous permettre d'étoffer un peu votre base de données afin de pouvoir l'utiliser lorsque le TP abordera la partie Requêtes d'interrogation.

A. Réfléchir pour modifier le modèle relationnel de la base de données

On souhaite ajouter dans la base précédente des données relatives à des cours et à des notes d'étudiants, sachant qu'un étudiant peut suivre plusieurs cours et obtenir plusieurs notes (note de CC, note d'Examen à la première session, note d'Examen à la deuxième session) pour un même cours. Pour chaque cours, un étudiant a un seul triplet de notes (note de CC, note d'Examen 1^{ère} Session, note d'Examen 2^{ème} Session⁷).

⁷ Cette note peut ne pas avoir de valeur si l'étudiant n'a pas passé de 2^{ème} session pour l'enseignement concerné.

QUESTION 18 : Parmi les choix ci-dessous, quel est à votre avis le "meilleur" modèle relationnel pour répondre aux besoins du paragraphe précédent ? Expliquer les avantages et inconvénients, lorsqu'il y en a, de chaque solution proposée.

1. Mettre toutes les informations sur les cours (intitulé, niveau, enseignant, etc.) et sur les notes dans la relation *Etudiants*.
2. Créer une relation (table) *Cours* et mettre toutes les informations sur les notes dans la relation *Etudiants* et lier la relation *Etudiants* à la relation *Cours*.
3. Créer deux relations (tables) *Cours* et *Notes* et lier la relation (table) *Notes* aux deux autres relations *Cours* et *Etudiants*.

B. Implémentation du nouveau schéma de base de données

Pensez à bien tester chaque attribut saisi!

1. Créer une nouvelle relation (table), intitulée *Cours*, dont le schéma est le suivant :
 - Un attribut *Numero*, clé primaire, qui est un nombre s'incrémentant automatiquement,
 - Un attribut *Intitule*, de type *Texte*, de taille 30, n'acceptant ni les valeurs *null* ni les chaînes vides,
 - Un attribut *Enseignant*, de type *Texte* (on supposera qu'il y a un seul enseignant pour chaque cours, pour simplifier), de taille 20 et dont les valeurs seront affichées entièrement en majuscules.
 - ① Cet attribut acceptera les valeurs *null* et les chaînes vides pour permettre, par exemple, d'insérer un cours dans la base de données dont on ne connaît pas encore l'enseignant.
 - Un attribut *Formation*, de type *Texte*, de taille 12 et dont la valeur pourra être choisie dans une liste de valeurs : "DU MI2E", "Licence MI2E" et "Master MIDO" par exemple.

QUESTION 19 : Si on avait voulu que plusieurs enseignants (ex. les enseignants d'Amphi, les chargés de TD etc.) apparaissent dans la base de données pour un même cours, quel modèle relationnel aurait-il fallu proposer ?

2. Créer une nouvelle relation (table), intitulée *Notes*, dont le schéma est le suivant :
 - Un attribut *RefEtudiant*, ayant les mêmes propriétés que l'attribut *NumeroCarte* de la relation *Etudiants*,
 - Un attribut *RefCours*, ayant les mêmes propriétés que l'attribut *N°* de la relation *Cours*, mais indexé avec doublons,
 - ① Attention, le type de ces deux attributs ne peut pas être *NumeroAuto*, car il sera clé étrangère (faisant référence à l'attribut *Numero* de la relation *Cours*), vous devez le choisir de type *Numérique* et de format *Entier long*.
 - Une clé primaire composée des deux attributs *RefEtudiant* et *RefCours*.
 - Les attributs *RefEtudiant* et *RefCours* sont indexés avec doublons.

① **Attention à ne pas créer trop d'index car ils ralentissent les insertions, les mises à jour et les suppressions de nuplets**⁸. Ici on crée deux index, car on prend comme hypothèses que les utilisateurs rechercheront souvent toutes les notes d'un étudiant ou toutes les notes d'un cours.

QUESTION 20 : Pourquoi la clé primaire est-elle composée de ces deux attributs et pas d'un seul des deux ? *Pour vous aider, relisez l'énoncé du problème*

- Un attribut *NoteCC*, de type *Numérique* prenant ses valeurs (que l'on suppose entières) entre 0 et 20, de valeur par défaut *null* (et donc acceptant les valeurs *null*) et de *Description* "Note de contrôle continu".

① Pour vérifier que la valeur de cet attribut est bien comprise entre 0 et 20, dans la propriété *Valide si*, saisir : ≥ 0 Et ≤ 20

Indiquer dans la propriété *Message si erreur* que "Les notes sont des entiers compris entre 0 et 20!".

- Un attribut *NoteExamSession1* de type *Numérique* prenant ses valeurs entières entre 0 et 20 et de valeur par défaut *null*⁹.
- Un attribut *NoteFinaleSession1* de type *Numérique*, prenant ses valeurs entières entre 0 et 20 et de valeur par défaut *null*. La valeur de cet attribut sera calculée à partir des valeurs des attributs *NoteCC* et *NoteExamSession1*, comme nous le verrons par la suite (voir item 5). Vous pouvez l'indiquer dans la *Description* de l'attribut. Notons que cette note finale n'est pas forcément entière.
- Un attribut *NoteExamSession2* de type *Numérique*, prenant ses valeurs entières entre 0 et 20 et de valeur par défaut *null*.

QUESTION 21 : Pourquoi indexe-t-on l'attribut *RefEtudiant* avec doublons ?

QUESTION 22 : Pourquoi indexe-t-on l'attribut *RefCours* avec doublons ?

QUESTION 23 : Quel est la "meilleure" *taille de champ* (celle qui occupera moins de place) pour les attributs *NoteCC*, *NoteExamSession1*, *NoteFinaleSession1* et *NoteExamSession2* ?

3. Lier les relations (table) *Notes*, *Cours* et *Etudiants*, de telle sorte que : (1) l'attribut *RefEtudiant* de la relation *Note* soit une clé étrangère qui fasse référence à l'attribut *NumeroCarte* de la relation *Etudiants* et (2) l'attribut *RefCours* de la relation *Note* soit une clé étrangère qui fasse référence à l'attribut *N°* de la relation *Cours*.

① Pour ajouter des tables dans la fenêtre *Relations*, cliquer sur le bouton  dans la barre d'outils en haut.



① N'oubliez pas de cocher l'intégrité référentielle.

Si le chargé de TD vous le demande, montrez-lui les relations que vous avez créées.

4. Insérer plusieurs nuplets dans les différentes relations (table) de la base de données.

⁸ La mise à jour des nuplets, l'insertion ou la suppression de nuplets impliquent une mise à jour des index correspondants.

⁹ Le fait que les attributs *NotesCC* etc. acceptent les valeurs *null* permet de ne pas avoir à saisir toutes les notes d'un seul coup, mais de laisser la possibilité à l'utilisateur de le faire en plusieurs étapes.

5. Pour introduire la feuille de TP suivante qui concerne les requêtes, vous allez modifier l'attribut *NoteFinaleSession1*, de telle sorte qu'il devienne un attribut calculé à partir des valeurs des attributs *NoteCC* et *NoteExamSession1*. Pour cela il va falloir créer une requête de mise à jour. Les étapes sont les suivantes :
- Dans le menu *Créer*, cliquer sur l'icône *Création de requête*.
 - Une fenêtre *Requête1* s'ouvre ainsi qu'une fenêtre permettant de sélectionner les relations (tables). Ajouter la relation *Notes*.
 - Fermer la fenêtre *Afficher la table* et cliquer sur l'icône *Mise à jour* dans la barre du haut.
 - Double-cliquer sur l'attribut *NoteFinaleSession1* dans la table apparaissant dans la fenêtre de votre requête – voir Figure 3. Le nom de l'attribut et celui de la relation (table) doivent apparaître dans la deuxième partie de la fenêtre (dans *Champ* et *Table*).
 - Dans *Mise à jour*, insérer la formule $([NoteCC]+[NoteExamSession1])/2$ indiquant que la valeur de l'attribut *NoteFinaleSession1* sera calculée¹⁰ en faisant la moyenne des valeurs des attributs *NoteCC* et *NoteExamSession1* – voir Figure 3.
 - Pour tester votre requête, cliquer sur le bouton . Un message doit s'afficher vous indiquant que vous allez mettre à jour plusieurs enregistrements. Si rien ne se passe, activer les requêtes de mise à jour en cliquant sur le message qui s'affiche au dessus de la fenêtre de la requête, puis recommencer.
 - Fermer la fenêtre de votre requête. Donner lui le nom *CalculNoteFinaleSession1*. Vous pourrez à tout moment exécuter à nouveau la requête en double-cliquant dessus (sur  *CalculNoteFinaleSession1*).
 - Ouvrir la relation (table) *Notes* pour vérifier les mises à jour effectuées.

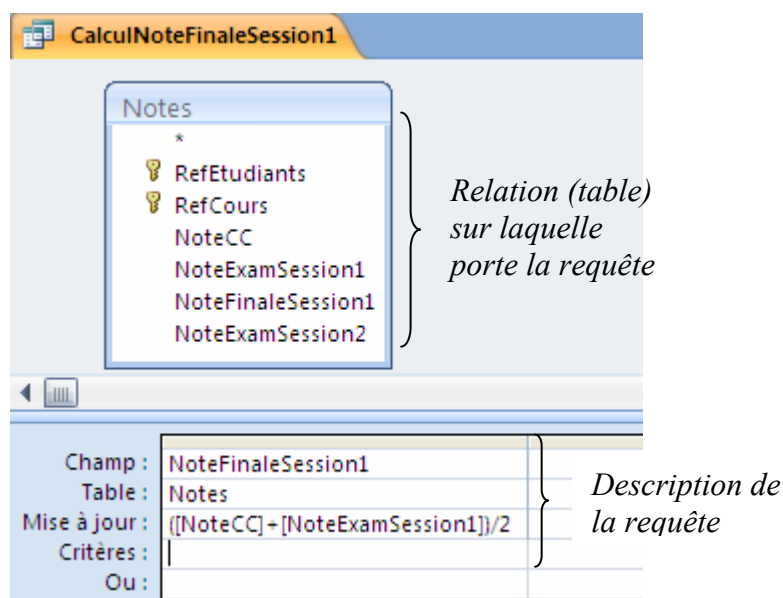


Figure 3 - Requête de mise à jour pour calculer la valeur d'un attribut.

- Retourner dans la Fenêtre de votre requête en *Mode Création*. Faire un clic droit sur le nom de la requête, et choisir *Mode SQL*. Vous voyez la requête que vous avez créée graphiquement, écrite en SQL, le langage standard des SGBD. Vous apprendrez ce

¹⁰ Ceci est un exercice, rien à voir avec le calcul de votre note finale de l'enseignement *Outils en Informatique* !!

langage dans la suite de votre formation (et également à l'inclure dans des programmes).

Si le chargé de TD vous le demande, montrez-lui votre requête de mise à jour et testez la devant lui

III. Pour finir le TP sur la création d'une base de données

Cette première feuille de TP a pour objectif principal de vous permettre de manipuler *Access* et de connaître les principales fonctionnalités pour créer une base de données.

Soyez curieux, après le TP ou pendant le TP si vous avez terminé cette feuille et que vous n'avez pas encore eu la suivante, faites d'autres tests. Vous pouvez créer des relations (tables) factices (dans une base de données *Test* – pour ne pas "*salir*" celle faite en TD) pour tester le format de certains types de données, certaines règles de validation, etc. Pensez à regarder les exemples dans l'aide. Plus vous manipulerez le logiciel, plus vous apprendrez...

Vous pouvez aussi regarder la base de données exemples d'*Access*, qui est stockée dans le fichier `ExempleBaseAccessComptoir.mdb` sur la page web :

<http://www.lamsade.dauphine.fr/~ozturk/courses.html/>